Российский государственный гуманитарный университет
Russian State University for the Humanities

RGGU BULLETIN

№ 12(55)/10

Scientific Journal

Information science. Information security.
Mathematics Series

Moscow
2010

ВЕСТНИК РГГУ

№ 12(55)/10

Научный журнал

Серия «Информатика. Защита информации.
Математика»

Москва
2010

УДК 94(560)
ББК 63.3(5)я54

ISSN 1998-6769

СОДЕРЖАНИЕ

Математические модели

Технологии

CONTENTS

Mathematical models

Technologies

От редакции

Во втором издании серии «Информатика. Защита информации. Математика» журнала «Вестник РГГУ» мы продолжаем публикацию материалов по актуальным аспектам сферы информационных технологий.

Главным событием этого года стало вступление в силу закона «О персональных данных», принятого в 2006 г. Целью закона является обеспечение защиты прав и свобод граждан при обработке их персональных данных, в том числе защиты прав на неприкосновенность частной жизни, личную и семейную тайну. Требование к операторам информационных систем персональных данных со стороны регуляторов, включающее такие механизмы госконтроля, как лицензирование деятельности по технической защите информации, сертификация средств защиты информации, подразумевает необходимое наличие достаточных материальных и трудовых ресурсов, которых у большинства операторов нет. Это привело к неготовности госорганов и органов местного самоуправления к исполнению закона с 1 января 2010 г. Подробнее положения закона рассматриваются в статьях данного выпуска.

Кроме того, уделено внимание таким вопросам, как качество обслуживания, технологии виртуализации, защита информационных систем, математические методы анализа систем, методы автоматического порождения гипотез и другим.

Д.А. Ларин

ЗАЩИТА ИНФОРМАЦИИ В ДРЕВНЕЙ РУСИ

В данной статье рассмотрен процесс зарождения методов защиты информации в нашей стране и первые факты их использования. Основное внимание уделено криптографическим методам защиты информации, приведены примеры древнерусских шифров. Статья охватывает временные рамки в основном с IX в. до конца XIV в.

Ключевые слова: тайнопись, криптография, шифр, стеганография.

С древних времен человечество стремится защищать свои тайны. Наиболее мощным средством защиты информации является криптография. Одними из первых криптографию стали использовать различные государственные структуры для защиты правительственных, военных, дипломатических сообщений. Здесь надо заметить, что при оценке влияния криптографии на те или иные события мировой истории следует использовать более широкое понятие – криптографическая деятельность. Под криптографической деятельностью понимается не только шифрование и дешифрование, но и организация каналов передачи сообщений (системы связи), использование различных методов защиты информации (криптография, стеганография, физическая защита собственных линий связи и т. д.), организация перехвата шифрованной информации противника. Дешифрование без перехвата невозможно. Разумеется, сюда входят меры по добыванию информации, облегчающей дешифрование (добывание ключей, описания шифрсистем и т. д.). С другой стороны, при разработке методов и средств защиты информации необходимо учитывать возможные аналогичные действия противника и предпринимать соответствующие меры для их пресечения. Если действия по добыванию информации связаны с разве-

дывательными операциями, то при защите главную роль играют контрразведывательные мероприятия. Поэтому криптографические службы работают в тесном контакте с разведкой и контрразведкой.

Вообще криптографическая деятельность является составной частью информационного противоборства, которое включает в себя организацию пропаганды и информационного воздействия на потенциального и реального противника и своего населения (поддержка патриотического духа, разъяснение политики государства и т. д.), ведение контрпропаганды, проведение операций по дезинформации противника. В случае проведения операций по информационному воздействию на противника нередко используется криптография. С одной стороны, узнав о дешифровании своих секретных сообщений, можно не усиливать защиту, а продолжать использовать тот же шифр, передавая дезинформацию, которую другая сторона будет принимать за истинную информацию. Такая ситуация называется дезинформацией «под шифром». В этом случае для передачи настоящей информации следует использовать другие шифры и другие каналы связи. С другой стороны, тайно захватив шифры и ключи противника (или вскрыв их аналитическим путем), можно попытаться от имени истинного отправителя передать противнику дезинформацию[1].

Фактически как только где-то происходило становление того или иного государства, как тут же начиналась криптографическая деятельность, с развитием государственных институтов учреждались специальные криптографические службы. Известный американский историк Дэвид Кан считает, что признаками великой державы являются наличие у страны ядерного оружия, успехов в освоении космоса и достижений в области криптографии[2].

Разумеется, не стала исключением и наша страна, история криптографической деятельности в России насчитывает не одно столетие. В данной статье рассказывается о событиях, которые привели к возникновению отечественной криптографической службы, которая сегодня является одной из лучших в мире. По словам Д. Кана, «русские вознесли достижения своей страны в сфере криптологии[3] до высоты полета космических спутников»[4].

В IX в. в результате объединения новгородских и киевских земель образовалось единое русское государство – Киевская Русь. Эффективное управление весьма обширной территорией было невозможно без организации надежной связи между столицей Киевом и подчиненными территориями и войсками, несущими сторожевую службу на границах Руси, а также находящимися в походе. Основным средством связи в то время были специальные княжеские гонцы «“люди пешие и конные” – и “верные головы” (люди из

княжеской дружины), которые со скоростью 200 и более верст в сутки передвигались от одного пункта до другого, передавая как устные, так и письменные сообщения»[5]. Использовались и другие способы связи: оптическая сигнализация с помощью костров и дымов, почтовые голуби, на поле боя управление осуществлялось с помощью сигнальных труб и свистков.

Для обеспечения конфиденциальности передаваемой информации использовали различные методы. Наиболее важные сообщения гонцы заучивали наизусть. При этом часто использовались намеки, иносказания, условные слова. Суть данного метода заключается в том, что смысл передаваемого сообщения мог понять только посвященный человек. Впоследствии в криптографии такой способ обеспечения секретности получил название «жаргонный код» и применяется до сих пор. Так, например, на жаргоне многих разведок слово *болеть* означает арест или заключение под стражу; *больница* – тюрьма; *доктор* – контрразведка. Тогда сообщение: «Майкл арестован контрразведкой. Ему грозит заключение в тюрьму» – принимает следующий «невинный» вид «Майкл заболел. Вчера был доктор и посоветовал ему лечиться в больнице». Использовался и так называемый «тарабарский язык»[6], когда в устное сообщение вставлялись частицы-паразиты. Так, фраза «возьми суму» звучала так: «тараВОбараЗЬМИтараСУбараМУ». Причем фраза произносилась как можно быстрее, человеку непривычному к такому способу общения, понять смысл сказанного было крайне затруднительно. Подобные способы защиты информации с древних времен были распространены не только у государственных служб, но и среди представителей криминального мира в различных странах, в том числе и в России.

Для защиты письменных сообщений применялись физическая защита, стеганография и шифрование. В качестве гонцов использовали физически крепких людей, они были хорошо вооружены, нередко гонец следовал в сопровождении охраны. Сами письма скручивались в свитки, опечатываемые специальными печатями, на которых была надпись «дьнеслово», что переводится как «скрытое, тайное слово». Такие печати были у многих русских князей, в том числе и у Александра Невского[7].

Стеганографический метод заключался в запрятывании сообщений, депеши зашивались в одежду, помещались в подошвы и каблуки обуви и т. д. Об этом, в частности, говорилось в одной из древних новгородских грамот: «...человиком грамотку пришли таино...»[8]

Шифрование. К сожалению, шифрованные документы, содержащие информацию государственного характера, относящиеся к эпохе Древней Руси, пока не обнаружены. Однако сохранился ряд

памятников русской письменности, в которых имеются зашифрованные фрагменты. В основном это летописи и тексты религиозного содержания. В этих источниках тайнопись применяется не столько для обеспечения секретности, сколько для того, чтобы подчеркнуть важность того или иного фрагмента, а также увековечить имя автора или переписчика. Именно эти документы дают возможность описать древнерусские системы шифрования.

С древних времен до конца XVII в. основным средством шифрования на Руси были различные варианты шифра простой замены (получившие название «иные письмена»), их можно разделить на три группы.

1. Замена знаков основного русского алфавита – кириллицы греческими буквами, а позднее латинскими. Также была распространена замена на буквы мало распространенных на Руси алфавитов глаголицы (рис. 1) и пермской азбуки (рис. 2). Эта азбука была создана епископом Стефаном Пермским в 1372 г. на основе кириллицы, греческого алфавита и древнепермских рунических символов. Некоторое время пермская азбука применялась на северо-востоке европейской части России, но широкого распространения не получила и примерно с XV в. использовалась как тайнопись. Что касается глаголицы, то в XI–XIII вв. глаголицей писали отдельные «наиболее секретные» слова, а в XIV–XVI вв. глаголицей записывались целые фразы и абзацы.

2. Замена знаков открытого текста на специально придуманные обозначения. На рис. 3 приведен пример ключа подобного шифра. Здесь написание ряда букв видоизменено, а для некоторых придуманы специальные обозначения. Подобные шифры на Руси назывались «измененными знаками». Этот способ тайнописи в Греции (откуда предположительно он пришел на Русь) назывался тахиграфией, он представляет собой изменение начертаний букв, когда писалась или часть буквы, или, наоборот, ее написание дополнялось новыми элементами. Часто тахиграфия совмещалась с использованием иностранных алфавитов. Использовалась для тайнописи и такая технология, как монокондил, или по-другому лигатура. Суть этого метода заключается в соединении при написании нескольких знаков-букв в одно целое. Для увеличения стойкости подобных шифров сообщение иногда записывали справа налево или вверх ногами. Практиковалось также видоизменение знаков письма, которое получило название «вязь». Надо отметить, что подобные системы шифрования часто сочетали в себе криптографию и стеганографию. Причудливые завитушки, вязь, различные выдуманные знаки противник мог принять за бессмысленные каракули, рисунки и т. д., а никак не за осмысленный текст.

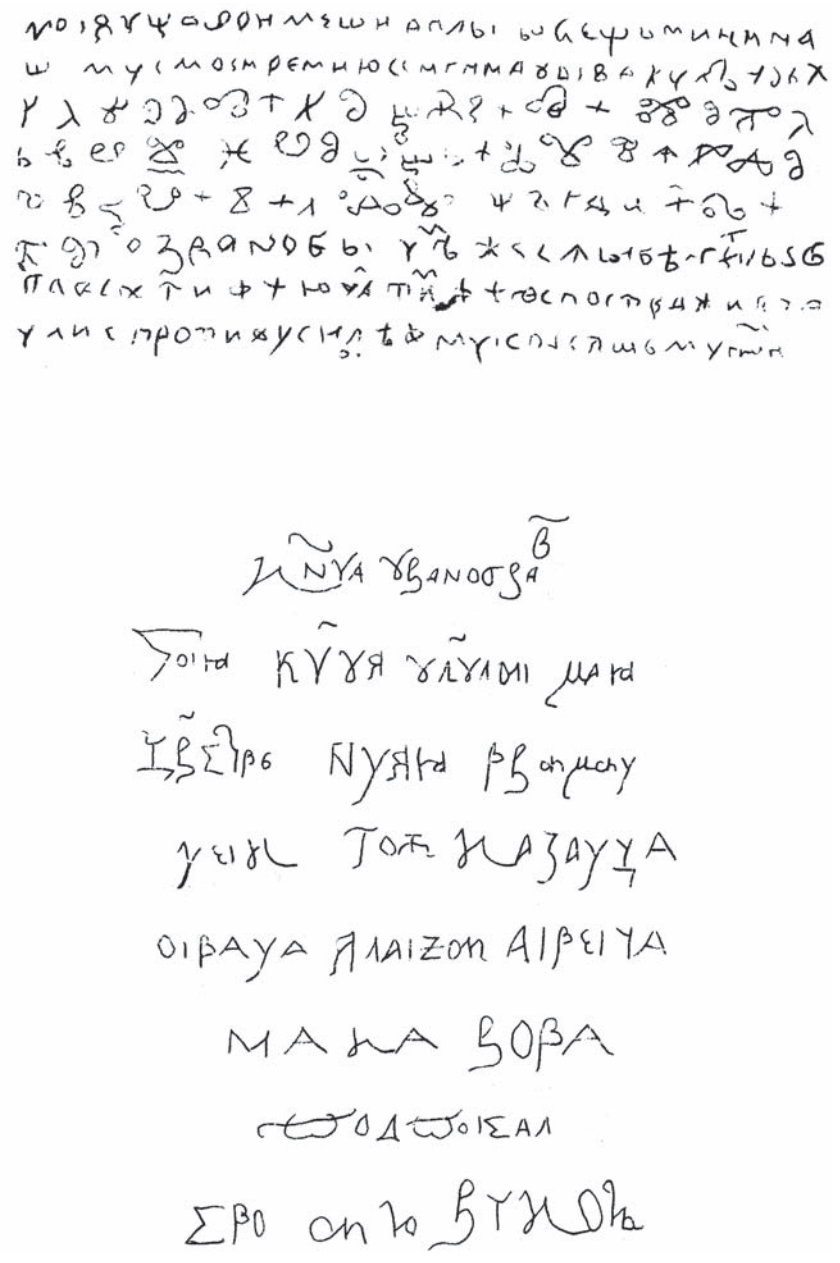


Рис. 1. Текст, написанный глаголицей[9]

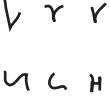
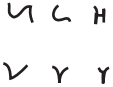
Русские буквы	Пермские письмена	Русские буквы	Пермские письмена
А		О	
Б		П	
Г		Р	
Д		С	
Е		Т	
Ж		У,В	
ДЖ		Ц	
З		Ч	
ДЗ		Ш	
И		Ы	
К		Ѣ	
Л		Ю	
М		Ӧ	
Н		Я	

Рис. 2. Пермская азбука[10]

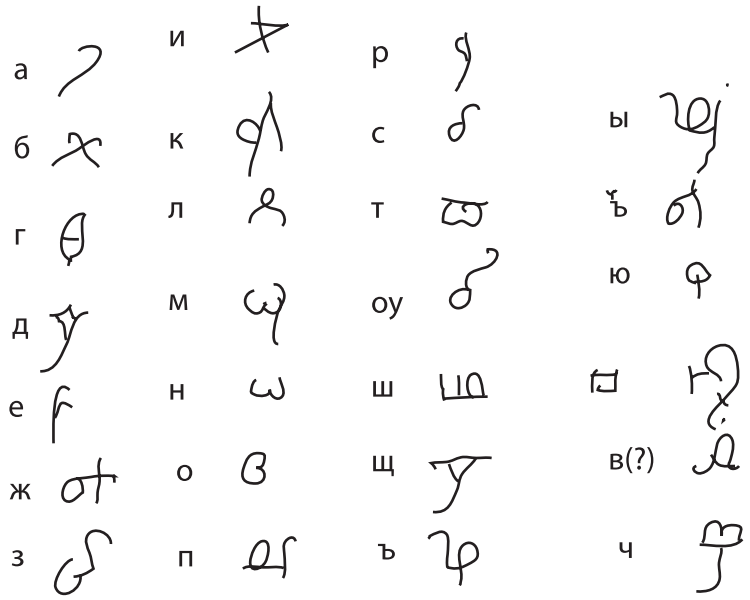


Рис. 3. Ключ к шифру простой замены (1590 г.)[11]

3. Геометрические системы простой замены, или, как их тогда называли, шифры в квадратах. На рис. 4 приведен открытый текст подобной системы, а на рис. 5 шифрованный, вместе они составляют ключ к шифру. Позднее такие системы получили название «масонский ключ».

Все эти виды тайнописи применялись на Руси, однако основной русской системой шифрования стал шифр простой замены, называвшийся «литорея». Существовало две разновидности этого шифра – простая литорея и мудрая. Простая литорея предполагает замену большинства согласных букв кириллицы на другие, взятые из того же алфавита, причем замену производят по определенному правилу. Гласные буквы и некоторые согласные (Й, Ь, Ъ) остаются без изменений. Самым простым и распространенным был следующий ключ. Выписывали в строку подряд все согласные в количестве 10 букв. Под этой строкой составляли вторую из последующих 10 согласных, но их записывали в обратном порядке, то есть справа налево:

Б	В	Г	Д	Ж	З	К	Л	М	Н
Щ	Ш	Ч	Ц	Х	Ф	Т	С	Р	П

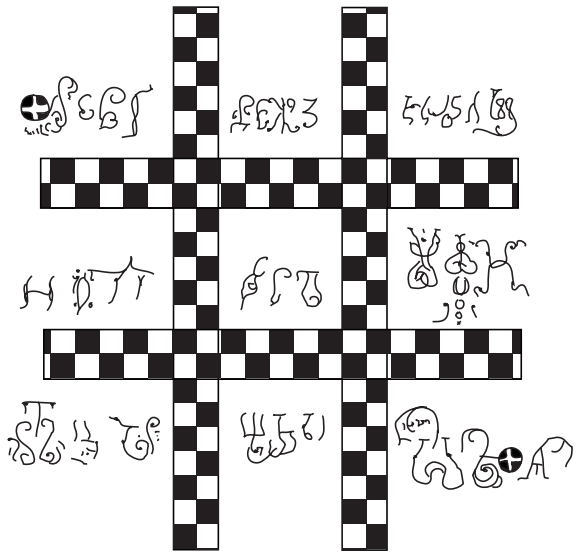


Рис. 4. Открытый текст геометрической системы простой замены (вторая половина XVII в.)[12]

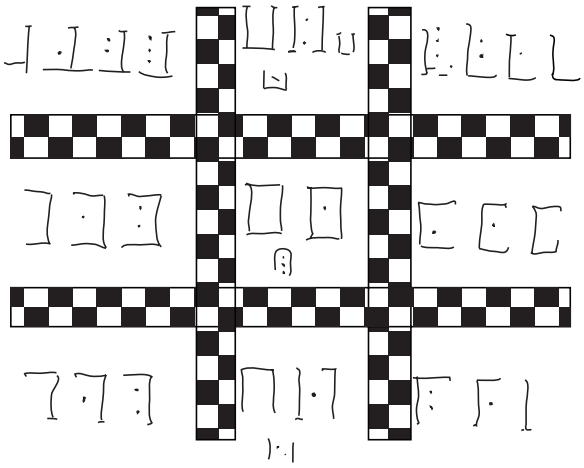


Рис. 5. Шифрованный текст геометрической системы простой замены (вторая половина XVII в.)[13]

Замена производилась следующим образом: буква из верхней строки заменялась на соответствовавшую ей букву из нижней строки и наоборот. Расшифрование осуществляется таким же образом. Для примера воспользуемся фрагментом Ермолинской летописи, датируемой 1463 г. В нем рассказывается о некоем жестоком княжеском слуге: «А прочих его чудес великое множество, невозможно ни описать, ни исчесть – потому что он во плоти есть ЦЬЯШОС (выделено авт. – *Д. Л.*)». Расшифровывая последнее слово с использованием вышеприведенного ключа, получаем: ДЬЯВОЛ. Летописец, очевидно, монах, зашифровал это слово, чтобы не поминать нечистого всуе.

Самый древний документ, известный на сегодня, содержащий зашифрованный простой литореей фрагмент, датируется 1229 г. Наиболее широкое распространение этот шифр приобрел в XIV–XV вв., но эта шифросистема использовалась и в XVIII в. Существовали усложнения простой литореи, когда буквы верхней и нижней строк располагались в случайном порядке. В мудрой литорее замене подлежали все буквы алфавита – и гласные, и согласные, следовательно, эта система представляет собой классический шифр простой замены.

Также на Руси были распространены цифровые, или счетные, системы шифрования, еще их называли «фиоть» или «хвиоть». Цифры в древней Руси записывались с помощью букв кириллицы и греческого алфавита (рис. 6). Чтобы указать, что буква означает цифру или число, а не звук, над ней ставили особый знак ~, который назывался «титло».

Система основана на использовании определенных букв кириллицы, имеющих известное цифровое значение. Такие буквы-цифры для тайнописи раскладывали на слагаемые, обычно «раздваивая», в результате чего вместо одной буквы-цифры записывали две. Например, буква «Д» имела значение «4», а после «раздвоения» записывалась двумя буквами, каждая из которых была равна половине преобразуемой буквы – писали рядом две буквы «ВВ», т. е. «22». Так поступали при зашифровке четных букв-цифр. Для нечетных применяли пары приближенных половинок. Например, вместо буквы «Е» (5) писали «ГВ» (т. е. 3+2). При этом буквы, не имеющие числового значения, не шифровались. Впоследствии числовое значение буквы стали разбивать на произвольные слагаемые, причем не на два, а на большее количество. Так, например, буква S (6) помимо традиционного ГГ могла передаваться как АЕ (1+5), АВГ (1+2+3) и т. д. В результате получается простейшая система многозначной замены. Очевидно, что чем больше числовое значение буквы, тем большим количеством вариантов она может быть представлена в шифрованном тексте.

Рис. 6. Буквенные обозначения цифр и чисел[14]

С помощью такого шифра увековечил свое имя в 1307 г. писец Домид, приписав к написанной в Пскове книге «Апостол» шифрованный текст (для удобства титло над буквами-цифрами опущено, а незашифрованное начало фразы переведено на современный русский язык): «а писал ВВ.МЛ.КК.ДД.ВВ.Ъ. ...рекше: двдъ, органъ, мысль, истина...» (последнее слово из-за порчи листа рукописи не читается)[15].

Ко второй части этой записи мы вернемся чуть позже, а пока расшифруем первую: ВВ=2+2=4=Д, МЛ=40+30=70=О, КК=20+20=М, ДД=4+4=8=И, ВВ=2+2=4=Д; таким образом, мы получаем имя писца «Апостола» 1307 г. – ДОМИДЪ.

Другим вариантом цифровой тайнописи являлся так называемый описательный. Примером его применения может служить текст XV в.: «Аще хощеши уведати имя писавшего книгу сию, и то ти написую: «Десятерица сугубая и пятерица четверицею и единъ; десятерица дващи и единъ, десятья четыре сугубо и четырежди по пяти; дващи два с единем; единица с четверицею сугубо; в семь имени словъ седмерица, три столпы и три души, царь». И сего числа в семь имени РОЕ»[16]. Расшифруем эту тайнопись. Слово «сугубо» в те времена было синонимом слова «дважды». Таким образом: «десятерица сугубая и пятерица четверицею» – это $10 \cdot 2 + 4 \cdot 5 = 40 = $ М; «единъ» – это 1 = А; «десятерица дващи» – $10 \cdot 2 = 20 = $ К; «единъ» – это опять 1 = А; «десятья четыре сугубо и четырежди по пяти»

$10 \cdot 4 \cdot 2 + 4 \cdot 5 = 100 =$ Р; «дващи два с единем» – $2 \cdot 2 + 1 = 5 =$ Е; «единица с четверицею сугубо» – $(1 + 4)2 = 10 =$ і (или в современной транскрипции Й). Тогда имя автора МАКАРЕЙ. Далее следует своеобразный проверочный код: букв в слове 7, «столпы и души» означают гласные и согласные, которых в слове по 3, а «царь» – это полугласная Й и, наконец, РОЕ – это 175, т. е. сумма буквцифр в слове.

Иногда шифрованные записи носили шутливый характер, для примера обратимся к обрывку берестяной грамоты XIII в., найденной в Новгороде[17]:

Н В Ж П С Н Д М К З Л Т С Ц Т...
Е Ъ Я И А Е У А А А Х О Е И А...

Если прочитать это сообщение по колонкам, то получаем следующий текст: НЕВЪЖЯПИСАНЕДУМАКАЗАЛХТОСЕЦИТА... Если адаптировать эту фразу к современному русскому языку и разделить слова пробелами, получим:

НЕВЕЖДА ПИСАЛ НЕ ДУМАВ СКАЗАЛ А КТО СИЕ ЧИТА(Л)... далее, очевидно, следовал нелестный отзыв об умственных способностях читателя. Для нас этот фрагмент важен тем, что это один из первых случаев применения на Руси шифра перестановки, пусть и в самой примитивной форме.

Самым древним способом сокрытия информации в текстах, использовавшимся на Руси, был акростих. Акростих (от греч. akros – крайний и stichos – строка) – стихотворение, в котором первые буквы строк образуют слово или фразу. Примером акростиха может служить стихотворение «Загадка акростишная» русского поэта Ю.А. Нелединского-Мелецкого (1752–1829):

Довольно именем известна я своим;
Равно клянется плут и непорочный им;
Утехой в бедствии всего бываю боле;
Жизнь сладостней при мне и в самой лучшей доле.
Блаженству чистых душ могу служить одна;
А меж злодеями – не быть я создана.

Если прочитать первые буквы строк сверху вниз, то получится «секретное» слово «дружба». Изобретателем акростиха считается известный древнегреческий комедиограф, философ и врач Эпихарм Сиракузский (примерно 550–460 гг. до н. э.). На Русь акростих пришел из Византии и начал широко применяться с XI в. Акростих на нашей земле имел множество наименований: начало-

строчие, началограние, краеграние, акростихиада, краестрочие, первобуквие и др. Акростих широко применялся поэтами, авторами и переписчиками книг, составителями надгробных эпитафий в основном для увековечивания своего имени. В.И. Даль называл акростих иместишием.

Применялся акростих и как средство защиты информации. Фактически применение акростиха можно отнести к стеганографическим методам защиты информации. На слух акростихи не воспринимаются. Чтобы их обнаружить, нужно читать написанное. Читающий заранее знает о существовании в тексте тайнописи, хотя в принципе может ее обнаружить самостоятельно. Древнерусские акростихи отличаются большим разнообразием. Известны многочисленные варианты тех или иных разновидностей акростихов. Так, чаще всего записи проявляются при чтении снизу вверх. Встречаются акростихи, в которых к обычной вертикальной записи добавляется целое слово или даже вся горизонтальная (верхняя или нижняя) строка, написанная открытым текстом, – это угловые акростихи. Можно сказать, что у наших книжников акростихи были излюбленным видом тайнописи. Посредством акростихов велась тайная деловая и личная переписка. В качестве примера приведем переписку двух религиозных деятелей – старцев Илариона и Феоктиста. Иларион в своем длиннейшем стихотворении первыми буквами строк выразил просьбу: «Старец господар Феоктист, даи ми книгу списат». Феоктист в ответ сочинил еще более длинное стихотворение, в котором таким же приемом составил ответ: «Старец господар Иларион, потруженная тобою любезне восприях и противу твоего, аще и не тако, но обаче, восписах ти вся, но ты же мя в том проси, никому не возвести».

Для защиты информации широко применялись «неправильные» акростихи. Такое название они получили потому, что записывались не только первыми буквами строк, но и двумя и более буквами, начинающими строки, и даже первыми слогами и словами строк. Ясно, что такая свобода действий существенно облегчала сочинителю составление весьма пространных тайнописных записей. Но вместе с тем такой произвольный подбор делает невозможным найти какую-то закономерность в чередовании читаемых букв, слогов и слов. Случайность чередования этих элементов исключает возможность правильно составить универсальный ключ. Иными словами, для каждой конкретной записи читающий должен иметь тот ключ, которым эта запись составлялась, т. е. это уже криптографическая система защиты информации.

Приведем пример «неправильного» акростиха. Иоанн Величковский (XVII в.) составлял различные тайнописи с именем Девы

Марии. Одно из них умещается в шести строках (имя «Мария» повторяется два раза и выделено заглавными буквами):

МАти блага,
РИза драга-
Я же нас крыет,
МАлодушных,
РИзо нужны,
Якъ руно греет.

Здесь выделенные буквы или слога в строках читают слева направо и сверху вниз. При этом первая буква третьей строки (Я) является не началом, а окончанием слова (драга/я).

Величковскому принадлежит и более сложная запись, когда буквы «секретного сообщения» располагаются в произвольном порядке и перемешаны между собой:

*М*ног*А*я **И**з не**СУ**щих **С**озда
сей
тво*Р*ен*ИЯ*
да*ДЕ*
ми **Х**е**Р**у**И**м**С**кую
ТОму
пе**С**нь х*ВА*ления.

Здесь зашифрованы два имени: «Мария Дева» (выделено заглавными буквами курсивом) и «Исус Христос» (выделено заглавными буквами и жирным шрифтом). Очевидно, что прочитать такую тайнопись, не зная ключа, крайне сложно.

Помимо стихотворений для сокрытия сообщений стали использоваться обычные тексты (принципы тайнописи те же, что для стихотворений) и даже бессмысленные наборы слов. Вернемся к шифрованной приписке к псковскому «Апостолу» 1307 г.: если взять первые буквы четырех последних слов, то получмм ДОМИ, последнее нечитаемое слово совершенно очевидно начиналось на Д, таким образом, и здесь получаем имя «Домид». Переписчик зашифровал свое имя двумя способами – цифровой тайнописью и акростихом.

Обеспечения секретности требовали следующие виды сообщений.

1. Информация, необходимая для организации государственного управления – княжеские указы и распоряжения, отчеты с мест и т. п. В процессе становления русской государственности система управления постоянно совершенствовалась. Киевскому князю под-

чинялись другие князья (в современном понимании – руководители регионов). Создавались службы, регулировавшие различные области человеческой деятельности (торговлю, ремесла, сбор налогов и податей и т. п.). Разумеется, в рамках деятельности по управлению государством постоянно возникала необходимость передачи секретной, тайной информации, и наиболее эффективными методами защиты правительственной информации являлись криптографические.

2. Военная информация. В первую очередь – передача «ратной вести, своевременное уведомление о появлении неприятеля, проведение сборов войск»[18], а также организация управления войсками во время боевых действий и в мирное время.

3. Дипломатическая информация. После становления древнерусского государства происходило налаживание связей с другими странами, в первую очередь с соседями. Основным внешнеполитическим партнером Древней Руси была Византия (хотя при этом русские князья неоднократно воевали с этим государством), дипломатические отношения периодически поддерживались также с рядом стран Европы, Хазарией и викингами. Примерами дипломатической деятельности Киевской Руси могут служить посольства к византийскому императору Феофилу II в 838 г. и королю Франции Людовику Благочестивому (839 г.). При этом следует отметить, что в те времена постоянных представителей в других государствах практически не было, послы отправлялись по мере необходимости. После крещения Руси князем Владимиром в 988 г. (отметим, кстати, что само это событие потребовало весьма серьезных дипломатических усилий) в дипломатической деятельности русского государства активно участвует Православная церковь. С усилением могущества древнерусского государства дипломатическая деятельность значительно активизировалась. Так, во время правления знаменитого князя Владимира Мономаха (1113–1125) и его сына Мстислава Великого (1125–1132) «...Русь поддерживала международные связи как с католическими, так и мусульманскими странами. Продолжали действовать союзные договоры с Венгрией и Польшей, были заключены династические браки с владетельными домами Швеции, Византии, Польши, Венгрии, Германии и других стран»[19]. Значительное место дипломаты Древней Руси уделяли разведывательной и другой тайной деятельности (заключение тайных союзов и соглашений, вербовка агентуры, подкуп чужих правителей, чиновников и т. д.). Подробнее о разведке чуть ниже. Вообще активная внешнеполитическая деятельность способствует развитию методов и средств защиты информации. Так как послам и другим дипломатическим работникам приходится работать на

территории иностранных государств (в том числе и потенциально или реально враждебных), то обеспечение конфиденциальности передаваемой на Родину информации приобретает крайне важное значение. Прав Д. Кан, утверждая, что «развитие криптографии находится в прямой зависимости от расцвета современной дипломатии»[20]. Важность работы криптографов оценили руководители многих государств.

4. Разведывательная информация. Важнейшим источником информации для любого государства является разведка. Разведывательную деятельность Киевской Руси можно разделить на стратегическую (агенты на территории иностранных государств, собирающие и передающие политическую и военную информацию) и тактическую (сторожевая служба, сообщающая о приближении неприятеля, и передовые отряды войска, ведущие разведку). Примером успешной работы стратегической разведки Древней Руси могут служить походы киевских князей на столицу Византии Константинополь, или, как у нас его называли, Царьград. В 860 г. русские дружины предприняли первый поход к столице Византии. Время нападения было выбрано крайне удачно. Византийская армия под предводительством императора Михаила III выдвинулась в Малую Азию для отражения нападения арабов, а флот ушел к острову Крит для борьбы с пиратами. Константинополь остался практически беззащитным. Осада города продолжалась неделю. Император Михаил III вынужден был срочно вернуться из Малой Азии и заключить мир с руссами. По условиям этого договора Византия выплатила большую контрибуцию и предоставила русским купцам выгодные условия для торговли на территории империи.

Не менее удачным оказался поход легендарного князя Олега, воспетого А.С. Пушкиным. В 907 г. он подошел к Константинополю в то время, когда мощный византийский флот отправился воевать с арабами, а полководец Андроник Дука поднял мятеж против центральной власти. Как и в предыдущий раз, дело кончилось контрибуцией и привилегиями русским купцам в Византии, помимо этого Византия обязалась выплачивать Руси ежегодную дань и содержать русских послов в Константинополе.

Летом 941 г. в поход на Византию отправился князь Игорь, и опять лучшие сухопутные части империи и ее флот находились далеко от Константинополя, воюя с арабами. К сожалению, в этот раз византийцы сумели получить информацию о нападении и организовать сопротивление. Войско Игоря понесло большие потери и вынуждено было отступить.

В 965 г. князь Святослав совершил поход против хазар, удар был нанесен в очень подходящий момент. Хазарский каганат был

истощен ожесточенной борьбой с Византией в Причерноморье и на Кавказе. Русские войска взяли штурмом столицу хазар Итиль, эта победа привела к распаду враждебного Руси государственного образования – Хазарского каганата.

Вышеперечисленные факты однозначно говорят, что русские князья были осведомлены о положении дел в стане врага и могли выбирать удобный момент для нападения. Что касается тактической разведки, то пограничная стража не раз предупреждала князей о нападении врагов, в основном это были кочевники – хазары, печенеги, половцы. О важности организации войсковой разведки в «Поучении» своим детям писал Владимир Мономах: «На войну вышед, не ленитеся, не зрите на воеводы; ни питью, ни едению не лагодите, ни спанью; и стороже сами наряживайте, и ночь, отовсюду нарядившие, около вои тоже лязите, а рано встанете; а оружья не снимайте с себе вборзе, не разглядевши ленощами, внезапу бо человек погыбаеть»[21]. Для ведения разведки русские князья назначали специальных людей из числа дружины. Основной задачей всех видов разведки было предупреждение об агрессии против государства и обеспечение подготовки собственного нападения на противника. Задачами тактической разведки были сбор сведений о противнике, разведка дорог, переправ, мест удобных для стоянки войска, наблюдение за противником на поле боя. В древнерусском языке разведчики назывались «прелагатаи», или «соглядатаи» (лица, засылаемые князем в стан врага, или завербованная агентура), и «просоки» (отдельные воины или небольшие отряды, следившие во время войны за вражеским войском). Передвижение своих войск прикрывалось специальными отрядами, получившими название «сторóжа». Разумеется, передача агентурной информации, сообщений с границ, а также информации от войсковой разведки требовала обеспечения их конфиденциальности, в том числе при помощи криптографических методов.

Кстати оригинальный способ передачи агентурной информации имел место в 988 г. при осаде князем Владимиром греческого города Херсонес. Войско Владимира осадило хорошо укрепленный город, но греки оказывали ожесточенное сопротивление, и осада могла продлиться долго. Но среди жителей Херсонеса нашелся сочувствующий русским человек по имени Анастас, он пустил к русскому войску стрелу, к которой была прикреплена записка. Она была опубликована знаменитым отечественным историком Н.М. Карамзиным и гласила: «...за вами, к востоку, находятся колодези, дающие воду херсонцам чрез подземные трубы; вы можете отнять ее». Как пишет далее Карамзин, «...великий князь поспешил воспользоваться советом и велел перекопать водопроводы... Тогда граждане, изнуряемые жаждою, сдались россиянам»[22].

А вот пример использования в Древней Руси такого способа информационной войны, как дезинформация. В 997 г. печенеги осадили один из русских городов, в этот момент князь с войском отсутствовал. Враги надеялись, что в городе закончится еда и его жители капитулируют. Осажденных спасла, как пишет Карамзин, «хитрость умного старца». Этот человек «велел ископать два колодезя, поставить в них одну кадь с сытою (медовый взвар на воде), другую с тестом и звать старшин неприятельских будто бы для переговоров. Видя сии колодези, они (враги. – *Д. Л.*) поверили, что земля сама собою производит там вкусную для людей пищу, и возвратились к своим князьям с вестию, что город не может иметь недостатка в съестных припасах»[23]. Получив такую информацию, печенеги отказались от продолжения осады.

Расширение территории Древнерусского государства, а также развитие системы государственного управления требовали совершенствования системы связи. В 920 г. повелением киевского князя был учрежден повоз – своеобразный прообраз современной почты. «Повозная повинность обязывала весь люд (кроме духовенства и бояр) по первому требованию княжеских людей (и в первую очередь гонцов) предоставлять им безвозмездно лошадей, корм, повозки и даже лодки с гребцами, что способствовало ускорению передачи государственных и ратных вестей»[24]. Вообще следует отметить, что первая треть XII в. стала для Руси периодом наивысшего расцвета политической и военной мощи, экономики, науки и культуры государства.

К сожалению, в XII в. из-за амбиций князей, роста экономического и политического могущества различных областей Киевской Руси и последовавшей затем междоусобной борьбы произошел разрыв политического и территориального единства Древнерусского государства, которое в конце концов распалось на ряд самостоятельных земель и княжеств. На Руси наступил период феодальной раздробленности. Отдельные княжества фактически превратились в независимые государства, которые проводили самостоятельную внешнюю и оборонную политику. Дипломатическая деятельность стала включать не только сношения с зарубежными государствами, но и отношения с соседними русскими княжествами. В каждом княжестве создавался свой государственный аппарат. Появлялись люди и службы, отвечающие за княжеский двор, хозяйство, финансы, суд, войско и т. п. В крупных княжествах (Киевское, Новгородское, Владимирское и др.) появились собственные «дипломатические службы»: при князьях состоял целый штат переводчиков, которых тогда называли толмачами, и дипломатов, способных отстаивать интересы своих земель при осуществлении посольств и приема иноземных делегаций у себя, имелась также и служба разведки. Система связи была хорошо поставле-

на во многих княжествах, что позволяло уже при приближении противника к границам княжества готовить войска к отражению нападения. Однако скорость передачи государственной и военной информации была слишком медленной. Требовалась постоянная, надежная, более быстро действующая связь.

В начале XIII в. на смену повозу пришла ямская гоньба, которая представляла собой конную эстафету и была организована в каждом княжестве. В результате договоренностей между князьями ямская гоньба обеспечивала связь на территории всей Руси. В начальный период монголо-татарского нашествия этот способ связи использовался для координации действий русских князей. Ямская гоньба как эффективное средство для передачи секретной информации сыграла важнейшую роль накануне и во время сражений, ставших поворотными пунктами нашей истории.

В 1240 г. новгородский князь Александр Ярославич, после известных событий получивший почетное прозвище Невский, организовал морские дозоры в Финском заливе и на реке Неве с целью предупреждения нападения с Запада. В начале июля 1240 г. один из дозоров обнаружил в устье Невы шведские корабли, выяснив место высадки и силы врага, с помощью ямской гоньбы послал тайное сообщение в Новгород: «уведав силу ратных, иде против князя Александра, да скажеть ему станы»[25]. 15 июля 1240 г. в месте впадения реки Ижоры в Неву произошла знаменитая Невская битва, в которой князь Александр наголову разгромил шведских агрессоров.

После поражения на Неве шведы прекратили на время атаки на Русь, но Новгородское княжество постоянно пробовали на прочность немецкие рыцари-крестоносцы, окопавшиеся в Прибалтике. После захвата немцами Пскова для Новгорода возникала реальная угроза со стороны крестоносцев. В начале 1242 г. Александр Невский начал собирать войска для сражения с Тевтонским орденом, и опять тайные вести о сборе войск передавались ямской гоньбой. Внезапным ударом Александр освободил Псков, однако главные силы крестоносцев еще не вступили в бой. Хорошо поставленная разведка и надлежащим образом сработавшая система связи (ямская гоньба) предоставили Александру Невскому информацию о противнике, что позволило выработать наилучшую тактику предстоящего сражения. Оно состоялось 5 апреля 1242 г. на льду Чудского озера и получило название «Ледовое побоище».

Характерным примером использования условной сигнализации служит случай, произошедший в 1261 г., когда татарский военачальник Бурундай с войском явился на территорию Галицко-Волынского княжества, расположенного на юго-западе русских земель. Бурундай предъявил князю Даниилу ультиматум: «Если вы со мною мир-

ны, то размечите (т. е. было велено разрушить укрепления. – *Д. Л.*) все свои города»[26]. Чтобы избежать татарского разорения, князь Даниил вынужден был подчиниться, укрепления большинства городов были разрушены. Но когда Бурундай и Василько, брат князя Даниила, подошли городу Холм, «любимому городу короля[27] Даниила»[28], то выяснилось, что жители города хорошо подготовились к обороне и не собирались уступать татарам. Бурундай понял, что взять город теми силами, что есть у него, не получится. «Тогда он обратился к Василько: “Этот город брата твоего, ступай, скажи гражданам, чтоб сдались”. С Василько Бурундай отправил трех татар и толмача, чтобы слушал, что князь говорит горожанам. Но Василько не растерялся: набрал каменей в руки и, подъехав к стенам, стал кричать холмским боярам: “Константин холоп, и ты другой холоп, Лука Иваныч! Это город брата моего и мой, сдавайтесь!” Сказав это, Василько три раза ударил камнем о землю (это был условный сигнал не верить его словам. – *Д. Л.*), давая этим понять, чтобы не сдавались, а бились с татарами. Боярин Константин все понял и отвечал Васильку: “Ступай прочь, если не хочешь, чтоб ударили тебя камнем в лицо, ты уже не брат королю, а враг ему”. Татары, бывшие с Василько, все пересказали Бурундаю (про условный знак они ничего не знали и не обратили на бросание камней внимания. – *Д. Л.*), и тот ушел от Холма пограбить Польшу, а оттуда возвратился в степи»[29].

Важнейшую роль система разведки и связи сыграла во время подготовки Куликовской битвы. Московский князь Дмитрий Иванович еще в начале 1370-х годов организовал целую систему защиты границы Московского государства (проходившую тогда по реке Оке) от нападений ордынцев. Она включала в себя «сторожи крепкие», «заставы», которые выдвигались далеко в степь для наблюдения за передвижениями ордынских войск, пограничные крепости (Коломна, Серпухов и др.), а также организацию связи с этими отрядами при помощи особых посыльных, а с крепостями при помощи ямской гоньбы. Для быстрой и эффективной мобилизации войска Дмитрий приказал учредить разрядные книги, «подробные росписи полков и воевод, благодаря которым были точно известны районы мобилизации, а также состав и численность участников похода»[30].

23 июля 1380 г. в Москву прискакал «сторож крепкий» Андрей Попов, сын Семенов, и сообщил: «Идет на тебя, государь, царь Мамай со всеми силами ордынскими, а ныне он на реке на Воронеже»[31]. После получения этой информации по всей Руси по ямским трактам полетели гонцы с секретными грамотами московского князя о сборе войск для отражения ордынского нашествия. С целью выяснить дальнейшие планы врага, выиграть время для сбора войск князь Дмитрий послал к Мамаю «юношу, доволна суща разумом и

смыслом, имянем Захарию Тютьшова»[32]. Захарий должен был вручить Мамаю большие дары и начать переговоры с ордынцами, пытаясь убедить их в покорности московского князя хану. Для переговоров вместе с Тютчевым были посланы «два толмача, умеюща языкъ половетцискый»[33]. По пути в Орду Тютчев получил информацию об измене рязанского князя Олега и союзе с Мамаем литовского князя Ягайло. Эта важнейшая информация была доведена до князя Дмитрия, Захарий «ославъ скоро вестника тайно к великому князю»[34]. Отметим, что Тютчев продолжил путь в Орду и прибыл к Мамаю. Дары и уверения в покорности московского князя были отвергнуты ханом, за смелые и достойные ответы на резкие выпады Мамая в отношении князя Дмитрия Тютчев был чуть не убит, но благодаря хитрости Захарию удалось избежать смерти и вернуться на Русь.

Получив сообщение об измене Олега и союзе Ягайло с Мамаем, князь Дмитрий решает нанести удар по Мамаю до соединения его полчищ с войсками Олега и Ягайло. Тем временем передовая стража продолжала наблюдение за основными силами Мамая, подтверждая, что хан не торопится идти на Москву, кочует у реки Воронеж, ожидая подхода союзников. В середине августа 1380 г. русские войска собрались в Коломне и оттуда форсированным маршем двинулись навстречу врагу. Движение войска сопровождалось постоянным получением информации о противнике от «сторожи» и специально высланных для разведки передовых отрядов. Благодаря грамотной организации разведки и связи князю Дмитрию с войском удалось раньше Мамая прибыть на место предполагаемого сражения и занять более выгодную тактическую позицию, а также не допустить соединения ордынцев с войсками Ягайло и Олега. 8 сентября 1380 г. на Куликовом поле состоялось знаменитое сражение, которое закончилось полной победой русского войска.

Одним из немногих общественно-политических документов, где имеются шифрованные фрагменты, является «Второе послание митрополита Киприана[35] 23 июня 1378 года». После кончины митрополита Алексия Киприан стремился занять высокий духовный пост в Москве. Против Киприана выступал московский князь Дмитрий Иванович. Киприан искал поддержки у игумена Троицкого монастыря Сергия Радонежского и игумена Симонова монастыря Федора. Отправитель, очевидно, учитывал, что содержание послания могло стать известным князю. Во избежание неприятностей Киприан отдельные части текста зашифровал. Им были скрыты имя одного из адресатов (Сергия) и духовный сан другого (Федора), а также некоторые слова и предложения.

Этот документ интересен тем, что в нем обнаружено 8 мест, зашифрованных простой литореей, при этом использовались разные

ключи! Рассмотрим «Послание» подробнее. К сожалению, до нас дошли лишь копии (их также называют списками) «Послания». Всего их сохранилось четыре – Мясниковский (начало XV в., наиболее древний) и три более поздних – Основной, Чудовский и Барсовский (они датируются концом XV – началом XVI в.). Во всех списках имеются шифрованные записи. Больше всего тайнописи осталось в древнейшем списке – шесть фрагментов, в остальных – по две-четыре записи. Сколько же было зашифрованных мест в исходном тексте, написанном митрополитом Киприаном, неизвестно.

В Мясниковском и Чудовском списках простейшим ключом (пример которого приведен выше) сделана единственная запись: «шлея мули гелкъпору...» При расшифровании получаем: «Всея Руси честъному...» В двух других списках этот фрагмент приведен уже в расшифрованном виде.

В «Послании» имеются еще три записи, в которых применен тот же простейший ключ, однако в словах заменялись не все согласные, а лишь часть: «...игумену Семчию и ичурепу Федору... (т. е. игумену Сергию и игумену Федору)». Вторая запись: «...едипъруцмепъ л шари. Не укаисъля от шал...» (...единомудрен с вами. Не утаилося от вас...). И третья запись всего в одно слово: «мода» (рода). Ясно, что если в таком частично дешифрованном тексте произвести согласно простейшему ключу замену всех согласных, то вместо расшифрованного текста получится новая тайнопись. Для чтения нужен именно тот ключ, которым пользовался сочинитель тайнописи. Все эти три тайнописи содержатся только в одном, древнейшем, списке, а в более поздних они либо опущены полностью или частично, либо приведены в расшифрованном виде. В Барсовском списке «Послания» есть два отдельных слова, зашифрованных усложненным ключом: «вобдввсни» (неблагословенни) и «пбокдати» (прокляти). Наконец, в «Послании» приведены еще две шифрованные записи: «Одеюрееви мивропродиву» (Олексееви митрополиту) и «Дв оудушь отдумени» (Да будут отлучени). Заметим, что во второй из этих записей переписчик допустил несколько ошибок, написав «дщ вудушь отдумини». Ключ к этим двум записям настолько сложен, что ни один из четырех переписчиков не расшифровал их – они в зашифрованном виде сохранились во всех списках.

Столь подробное рассмотрение здесь зашифрованных литореей фрагментов в разных списках «Второго послания митрополита Киприана 23 июня 1378 года» нужно, чтобы показать, что, во-первых, применявшиеся для шифрования ключи даже в одном документе разнообразны. Во-вторых, переписчики вмешивались в текст тайнописи, в большинстве случаев его расшифровывали, иногда лишь

частично, а также при переписке допускали ошибки, затруднявшие понимание написанного, или вовсе отбрасывали непонятное.

Подведем некоторые итоги. В процессе становления и развития русского государства в IX–XIV вв. криптографическая деятельность играла весьма важную роль в организации системы государственного управления, передачи военной, дипломатической и разведывательной информации. Главным достижением данного периода является создание на Руси достаточно надежной и эффективной системы связи – повоза, а впоследствии ямской гоньбы. Из вышеперечисленных примеров можно сделать однозначный вывод, что ямская гоньба сыграла весьма существенную роль в процессе преодоления феодальной раздробленности Руси и создания Московского государства. Она была одним из ведущих средств государственного управления.

Для обеспечения секретности передаваемой информации использовались различные методы, в том числе криптографические, однако следует отметить, что время регулярных служб по защите информации еще не пришло. Мероприятия по защите информации проводились по мере необходимости, при этом выбор того или иного способа защиты информации осуществлял сам князь или довольно ограниченный круг его доверенных людей, они же и осуществляли подобные мероприятия, в том числе шифрование и расшифрование тайных сообщений. Не существовало каких-либо универсальных подходов к выбору методов и средств защиты информации, все зависело от конкретной ситуации, и в схожих случаях могли приниматься совершенно различные решения. С усилением русского государства, его военной и экономической мощи, расширения масштабов внешнеполитической деятельности потребовалось проведение мероприятий по защите информации на регулярной основе, понимание того, что этим должны заниматься специально подготовленные люди и службы.

Примечания

1 См.: *Гольев Ю.И., Ларин Д.А., Тришин А.Е., Шанкин Г.П.* Криптография: страницы истории тайных операций. М.: Гелиос АРВ, 2008.

2 См.: *Кан Д.* Война кодов и шифров. М.: РИПОЛ КЛАССИК, 2004.

3 На западе часто употребляется этот термин для обозначения науки о криптографических методах защиты информации. Криптология разделяется на криптографию (науку о создании шифров) и криптоанализ, изучающий методы их взлома. У нас в стране для обозначения данной сферы деятельности принят термин «криптография».

4 *Кан Д.* Указ. соч. С. 261.

[5] *Астрахан В.И., Гусев В.В., Павлов В.В., Чернявский Б.Г.* Становление и развитие правительственной связи в России. Орел: ВИПС, 1996. С. 10.

[6] Интересно отметить, что шифрованные записи в Древней Руси называли «тарабарской грамотой».

[7] *Соболева Т.А.* История шифровального дела в России. М.: ОЛМА-ПРЕСС-Образование, 2002. С. 43.

[8] *Бабаш А.В., Шанкин Г.П.* История криптографии. Ч. I. М.: Гелиос, 2002. С. 207.

[9] Там же. С. 203.

[10] См.: *Соболева Т.А.* Указ. соч.

[11] Там же. С. 30.

[12] Там же. С. 31.

[13] Там же. С. 32.

[14] *Бабаш А.В., Шанкин Г.П.* Указ. соч. С. 202.

[15] *Соболева Т.А.* Указ. соч. С. 37.

[16] *Бабаш А.В., Шанкин Г.П.* Указ. соч. С. 205.

[17] Там же.

[18] *Астрахан В.И., Гусев В.В., Павлов В.В., Чернявский Б.Г.* Указ. соч. С. 10.

[19] *Кудрявцев Н.А.* Государево Око. Тайная дипломатия и разведка на службе России. М.: ОЛМА-ПРЕСС, 2002. С. 37.

[20] Цит. по: *Бабаш А.В., Шанкин Г.П.* Указ. соч. С. 182.

[21] Цит. по: *Кудрявцев Н.А.* Указ. соч. С. 14–15.

[22] Цит. по: Очерки истории внешней разведки: В 5 т. / Под ред. Е.М. Примакова, С.Н. Лебедева. М.: Международные отношения, 1999. С. 17.

[23] Там же. С. 17.

[24] *Астрахан В.И., Гусев В.В., Павлов В.В., Чернявский Б.Г.* Указ. соч. С. 11.

[25] *Кудрявцев Н.А.* Указ. соч. С. 75.

[26] *Широкорад А.Б.* Дипломатия и войны русских князей. М.: Вече, 2006. С 162.

[27] В 1257 г. Папа Римский даровал Даниилу королевскую корону в надежде, что тот перейдет в католичество. Католиком Даниил так и не стал, но тем не менее Даниил и его потомки были единственной королевской династией на Руси. Подробнее см.: *Широкорад А.Б.* Указ. соч. С. 161–162.

[28] Там же. С. 163.

[29] Там же.

[30] *Кудрявцев Н.А.* Указ. соч. С. 109.

[31] Там же.

[32] Там же.

[33] Там же.

[34] Там же.

[35] Киприан (ок. 1336–1406), религиозный деятель, русский митрополит в 1380–1382 и с 1390 г., в 1382 г. руководил неудачной обороной Москвы от нашествия хана Тохтамыша. Активный сторонник сына Дмитрия Донского великого князя Василия I.

В.В. Белов, А.В. Некраха

ЗНАЧЕНИЕ ПАТЕНТНОЙ ИНФОРМАЦИИ ДЛЯ ОБЕСПЕЧЕНИЯ ИННОВАЦИОННОГО РАЗВИТИЯ СТРАНЫ

Цель данной статьи – анализ проблемы модернизации национальной системы патентной информации. На основе официальной статистики, опубликованной Всемирной организацией интеллектуальной собственности, установлена связь между качеством системы патентной информации и уровнем национальной экономики. Авторы проводят изучение информационных потребностей пользователей в патентной информации. В настоящее время патентная информация является наиболее эффективным источником знания о современной технологии. С этой точки зрения базы данных Патентного ведомства США и Европейского патентного ведомства представляют большой интерес. Рассматривается структура и основные принципы поиска в данных базах. Особое внимание авторы уделяют базам данных российского патентного ведомства. К сожалению, сравнение между национальной патентной системой и аналогичными системами развитых стран не в пользу национальной системы. Принимая во внимание этот факт, авторы выдвигают ряд предложений по модернизации систем патентной информации в нашей стране.

Ключевые слова: информация, патент, модернизация, патентное ведомство, патентный поиск, современная технология, источник информации.

Одной из наиболее актуальных проблем российской экономики сегодня является повышение конкурентоспособности отечественных товаров, работ и услуг. Это может быть достигнуто преимущественно за счет использования в них новейших результатов интеллектуальной деятельности, снижения себестоимости продукции благодаря применению ресурсосберегающих технологий. В обеспечении конкурентоспособности отечественной экономики первоочередную роль играет активизация инновационной деятельности. Од-

ной из основных задач государства является проведение мер, направленных на ее существенное оживление.

Развитые страны мира (США, Япония, Германия, Франция) добиваются более половины прироста ВВП за счет постоянного совершенствования и освоения новых технологий. Это обеспечивает им долговременные конкурентные преимущества и высокий уровень жизни. Доля же России на глобальном рынке наукоемкой продукции составляет менее 1%.

На рис. 1 показаны данные Евростата, иллюстрирующие соотношение организаций, вводящих инновации, в различных странах[1].

Безусловно, инновации базируются на изобретениях, которые, как правило, защищаются патентами.

Несмотря на то что во всех странах мира доля принятых изобретений составляет несколько процентов от подаваемых заявок на патент, количество таких заявок явным образом соотносится с инновационным уровнем страны.

На основе анализа патентно-информационных ресурсов можно сделать вывод об интенсивности научно-исследовательских работ по конкретным областям техники в различных странах. Существует прямая связь между количеством заявок на патент и научно-техническим уровнем страны в целом.

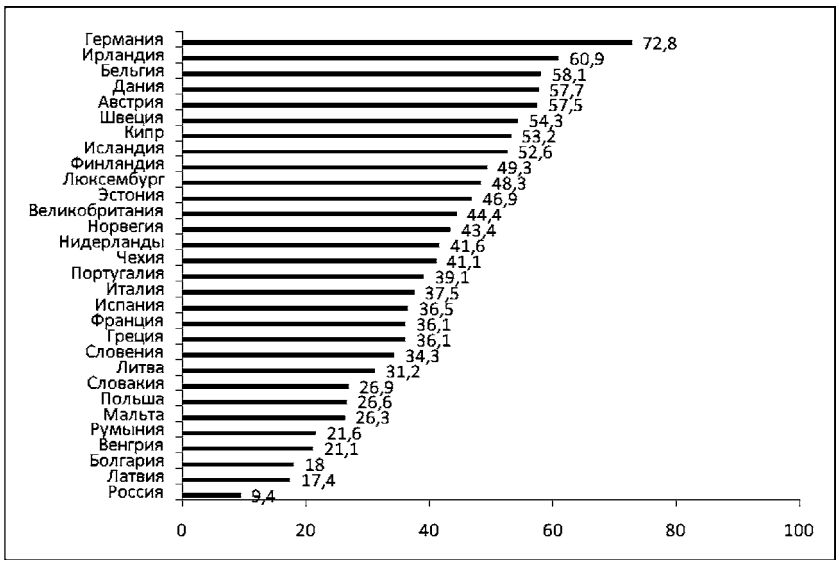


Рис. 1. Удельный вес организаций, осуществлявших технологические инновации, в общем числе организаций

В таблице приведены статистические данные по количеству заявок на патент, поданных по инновационно значимым отраслям техники за период с 2001 по 2005 г. Таблица построена по данным Всемирной организации интеллектуальной собственности[2].

Таблица

Количество поданных заявок на патент

Область техники	Страна					
	Япония	США	Германия	Франция	Китай	Россия
Телекоммуникации	206626	110999	20679	15681	66282	1885
Компьютерные технологии	256879	195085	26145	11707	45345	1522
Полупроводники	219804	70207	19165	4273	63186	746
Биотехнологии	32146	92204	17540	7274	6050	7083
Нанотехнологии	5177	4165	1353	487	1263	71
Фармацевтика	36521	134682	30887	14359	5290	6227

Из таблицы очевидно следует, что в России должны быть предприняты энергичные меры по преодолению существующего технологического отставания от ведущих стран мира. Например, количество заявок на изобретения в области нанотехнологий в Российской Федерации в 73 раза меньше, чем в Японии и 59 раз меньше, чем в США. Еще больший разрыв существует в области полупроводников. Здесь количество заявок в нашей стране меньше, чем в Японии в 295 раз (!) и в 94 раза меньше, чем в США.

Приведенные данные свидетельствуют о чрезвычайно низкой изобретательской активности в России и отсутствии мотивации к проведению инновационной политики.

Эта проблема осознается руководством страны, и в последние годы принят ряд мер для изменения ситуации. В 2006 г. Президентом Российской Федерации были утверждены «Приоритетные направления развития науки, технологий и техники Российской Федерации» и «Перечень критических технологий Российской Федерации».

В 2007 г. впервые за последние годы Министерством образования и науки был разработан проект долгосрочного прогноза научно-технологического развития Российской Федерации на период до 2025 г., сделана попытка определить место Российской Федерации в глобальном пространстве науки и технологий, выстроить национальные приоритеты научно-технологического развития.

В феврале 2008 г. были проведены парламентские слушания по вопросам инновационной политики. Среди рассмотренных вопросов: финансирование научно-исследовательских работ, развитие малого инновационного бизнеса, взаимодействие науки и производства, а также ряд других вопросов.

Однако, на наш взгляд, до настоящего времени не уделяется должного внимания проблеме информационного обеспечения, хотя это является неотъемлемой частью инновационного пути развития государства. Необходимой научной и технической информацией и современными информационными технологиями должны быть обеспечены все стадии инновационного цикла: фундаментальные исследования – поисковые НИР – прикладные НИР – опытно-конструкторские работы – освоение производства новой продукции и коммерциализация инноваций, а также образовательная сфера.

Важным элементом в обеспечении информационных потребностей субъектов инновационной деятельности России должна стать патентная информация. В современных условиях для успешной инновационной деятельности необходим исчерпывающий технический кругозор, который наилучшим образом может быть обеспечен только патентной информацией. Патентная информация во всем мире считается самой оперативной, достоверной и хорошо структурированной. Она появляется обычно раньше других сведений о каком-либо изобретении, так как большинство компаний предпочитает не раскрывать результаты своих исследований по очевидным причинам конкуренции. Но если какая-либо компания желает получить исключительные права на свое изобретение, она должна подать патентную заявку, которая, по всей вероятности, будет опубликована и станет доступной публике. Это объясняет, почему патенты являются первой публикацией по данному изобретению и наиболее актуальным источником информации по какой-либо технологии.

При разработке конкурентоспособных объектов техники и технологий необходимы знания об отечественных и зарубежных изобретениях и полезных моделях, а также других объектах промышленной собственности (ОПС). Такие сведения содержатся в патентно-информационных ресурсах, основу которых составляют патенты на ОПС. Патентно-информационные ресурсы используются практически на всех стадиях НИОКР для определения технического уровня и тенденций развития объектов техники или технологии, их патентоспособности, патентной чистоты и конкурентоспособности, а также инновационной политики.

Информация, содержащаяся в описаниях изобретений, полезных моделей и промышленных образцов, позволяет экономить значительные средства на проведение научно-исследовательских работ.

В соответствующем патентно-информационном обеспечении нуждаются все заинтересованные участники системы охраны промышленной собственности:

– организации и предприятия, создающие научно-технические новшества и способствующие вовлечению результатов научно-технической деятельности в хозяйственный оборот;

– патентовладельцы и оспаривающие их права оппоненты;

– судебные органы;

– патентные поверенные и другие.

Исходя из заявленной политики государства, направленной на переход от сырьевой экономики к инновационной, доступ к информации в области промышленной собственности должен быть повсеместным, удобным и бесплатным. Такие условия могут обеспечить только доступные через Интернет базы данных патентной информации. Это особенно актуально для нашей страны, учитывая ее существование в 11 часовых поясах. Практически единственным способом получения оперативной информации является обращение во Всемирную сеть. В настоящее время накоплен огромный опыт в использовании Интернета для целей информационного поиска патентных документов.

Наиболее часто используемыми для поиска патентной информации сайтами являются сайт Патентного ведомства США и сайт Европейского патентного ведомства (ЕПВ).

Сайт Патентного ведомства США, расположенный по адресу www.uspto.gov, предоставляет обширные массивы информации как о деятельности ведомства, так и по всем основным аспектам патентного дела и изобретательства. Совершенствованию сайта уделяется большое внимание, что выражается в динамичном развитии его структуры, расширении состава информации, форм ее подачи, способов интерактивного общения с пользователями.

Патентное ведомство США позиционирует свой информационный ресурс как общедоступный и рассчитанный на широкую публику. Доступ к базе данных и ее возможностям осуществляется на безвозмездной основе.

Патентная база США является одной из самых больших технических баз данных в мире, она насчитывает более 7 млн патентов, классифицированных по 145 тыс. типов изобретений. Ежемесячно к базе обращается 350 тыс. IP-адресов и просматривается около 36 млн факсимильных изображений патентов и 150 млн страниц.

Сайт содержит две автономные базы данных:

1) Issued Patents (PatFT) – патенты;

2) Published Applications (AppFT) – патентные заявки.

База PatFT представляет собой два больших информационных массива: полнотекстовые патенты начиная с 1976 г. (full-text patents since 1976) и факсимильные изображения патентов начиная с 1790 г. (full-page images since 1790). Патенты, зарегистрированные в период с 1709 по 1976 г. доступны для просмотра в графическом виде (TIF-формат), полученном путем сканирования патентной документации. Поиск таких патентов возможен только по номеру патента, дате регистрации и по коду национальной патентной классификации. Патенты, зарегистрированные после 1976 г., могут быть найдены при помощи всех возможностей поисковой системы.

Уникальные патентно-информационные массивы предоставляются в свободное пользование Европейским патентным ведомством.

В 1998 г. ЕПВ публиковало проект Распределенной патентно-информационной службы (DIPS), ориентированной преимущественно на широкую общественность и удовлетворение нужд индивидуальных пользователей, а также малых и средних предприятий.

Необходимость обеспечения свободного доступа к патентной информации была мотивирована тем, что она генерируется в ходе патентной процедуры, оплачиваемой за счет пошлин заявителей. Поэтому сегодня такая информация доступна всем пользователям патентной системы в бесплатном режиме, в каких бы целях она ни использовалась – личных или коммерческих, причем с оплатой только изготовления копий документов и их доставки. Это создало благоприятные условия не только для конечных пользователей, но и для информационной индустрии, которая сегодня использует данные ЕПВ для предоставления своим клиентам услуг, обогащенных дополнительными потребительскими свойствами, удерживая приемлемый уровень цен.

ЕПВ содержит информацию о патентных документах 72 стран в объеме более 60 млн единиц. Кроме того, возможен доступ к базам данных патентных документов ведомств стран – участников Европейской патентной конвенции. Весь информационный массив ЕПВ делится на два уровня. Первый уровень представляет собой серверы национальных патентных ведомств стран-участниц с глубиной предоставления информации минимум в 24 месяца, этот уровень называется «I esp@cenet».

Второй уровень называется «II esp@cenet», он представляет собой три базы данных: Worldwide, EP esp@cenet и WIPO esp@cenet.

EP esp@cenet – база данных, содержащая библиографические сведения (HTML-формат) и факсимильные копии (PDF-формат) патентных заявок EPO (European Patent Organisation), опубликованных в течение последних 24 месяцев (EP documents).

Worldwide – база данных, содержащая патенты (либо патентные заявки) более 70 национальных и нескольких международных патентных организаций.

Объем доступного материала различен для разных стран – от библиографических сведений (HTML-формат) до полных текстов (HTML-формат) и факсимильных копий (PDF-формат).

Временной охват тоже варьируется от страны к стране (от нескольких лет до десятков лет). На сайте можно получить абсолютно точную информацию по глубине предоставления и доступному переводу патентной информации по каждой стране в виде подробного отчета в PDF-формате.

WIPO esp@cenet – библиографические сведения (HTML-формат) и факсимильные копии (PDF-формат) патентных заявок (PCT-publications) международной патентной организации WIPO – World Intellectual Property Organization, опубликованных в течение последних 24 месяцев (WO documents).

Архитектура сайта позволяет локализовать интерфейс поиска для страны, патентное ведомство которого участвует в информационном обмене, эта функция возлагается на национальное патентное ведомство. Как правило, глубина перевода ограничивается элементами поиска и небольшими подсказками. Полный список стран и организаций, имеющих свой интерфейс, расположен по адресу http://www.espacenet.com/access/index.en.htm.

В нашей стране в июле 1999 г. Роспатент ввел в эксплуатацию свой информационный ресурс (www.rupto.ru), который предоставлял доступ к реферативной базе изобретений Российской Федерации начиная с 1994 г. В отличие от рассмотренных выше баз данных Патентного ведомства США и ЕПВ, изначальная политика Роспатента не предполагала бесплатного доступа ко всем информационным ресурсам. По этой причине в 2000 г. был открыт платный доступ к полнотекстовым базам данных изобретений и к библиографической базе данных товарных знаков начиная с 1991 г., а с 2002 г. – платный доступ к библиографической базе данных промышленных образцов, в том числе к их цветным изображениям. Со временем состав баз данных расширился, на данный момент весь массив информации, доступный для пользователей через Интернет, выглядит следующим образом.

RUPAT-БД содержит полные тексты российских патентов на изобретение и (частично) заявок на изобретение, графическую информацию. Состав БД соответствует составу официальных бюллетеней за 1994–2008 гг. Периодичность пополнения – три раза в месяц. Объем БД на апрель 2008 г. составил 356117 документов. Платный доступ.

RUABRU-БД содержит рефераты российских патентов на изобретение и формулы заявок на изобретение, графическую информацию – основной рисунок. Состав БД соответствует составу официальных бюллетеней за 1994–2008 гг. Периодичность пополнения – три раза в месяц. Объем БД на апрель 2008 г. составил 611 287 документов. Бесплатный доступ.

RUABEN-БД содержит рефераты на английском языке российских патентов на изобретение, графическую информацию – основной рисунок. Периодичность пополнения – три раза в месяц. Объем БД на апрель 2008 г. составил 355 160 документов. Бесплатный доступ.

RUPAT_OLD-БД содержит полные тексты российских патентных документов с 1924 до 1994 г. в факсимильном виде. В связи с автоматической обработкой патентных документов в цифровой формат точность поиска не гарантируется и в представленной библиографической информации возможны ошибки. Объем БД на апрель 2008 г. составил 1 432 095 документов. Платный доступ.

RUABU1-БД содержит полные тексты российских патентов на полезные модели, графическую информацию. Состав БД соответствует составу официальных бюллетеней за 1994–2008 г. Периодичность пополнения – три раза в месяц (описания пополняются один раз в три месяца). Объем БД на апрель 2008 г. составил 72 807 документов. Платный доступ.

Сопоставление характеристик баз данных и предоставляемого сервиса патентными ведомствами США, ЕПВ и Роспатента показывает, что российский изобретатель поставлен в неравные условия с его западными коллегами, которые могут беспрепятственно получать через Интернет подробную, удобную в доступе и бесплатную информацию о последних достижениях мировой науки, отраженную в патентных документах разных стран, в том числе и российских.

Сложившаяся ситуация противоречит не только логике развития инновационной деятельности как инструмента подъема экономики, но и ряду положений Федерального закона «Об информации, информационных технологиях и защите информации». В частности, в ст. 12 ч. 1 декларируется: «1. Государственное регулирование в сфере применения информационных технологий предусматривает:

1. Регулирование отношений, связанных с поиском, получением, передачей, производством и распространением информации с применением информационных технологий (информатизации), на основании принципов, установленных настоящим Федеральным законом.

2. Развитие информационных систем различного назначения для обеспечения граждан (физических лиц), организаций, государственных органов и органов местного самоуправления информацией, а также обеспечение взаимодействия таких систем.

3. Создание условий для эффективного использования в Российской Федерации информационно-телекоммуникационных сетей, в том числе сети Интернет и иных подобных информационно-телекоммуникационных сетей».

В настоящее время эффективность использования информационно-телекоммуникационных сетей, в том числе сети Интернет, в отношении баз данных ОПС недостаточна. Причем это не вопрос технической возможности, а вопрос, видимо, сознательно выбранного ограничения. Для изменения ситуации необходимо вводить более четкие правовые формулировки, соответствующие курсу государства, который достаточно полно отражен в документе «Стратегия развития информационного общества в Российской Федерации», где говорится: «Международный опыт показывает, что высокие технологии, в том числе информационные и телекоммуникационные, уже стали локомотивом социально-экономического развития многих стран мира, а обеспечение гарантированного свободного доступа граждан к информации – одной из важнейших задач государств».

В результате проведения сравнительных исследований доступа к патентной информации можно заключить, что ситуация с доступом к патентной документации Роспатента по средствам Интернет требует серьезных изменений для соответствия как требованиям времени, так и нормативным документам.

С правовой точки зрения представляется целесообразным выделить патентную информацию, размещаемую государственными органами в информационно-телекоммуникационных сетях, в отдельный вид информации, которая предоставляется в бесплатный доступ, включая пользование поисковым аппаратом. Такое дополнение можно внести в ст. 8 Федерального закона «Об информации, информационных технологиях и о защите информации» отдельным пунктом.

Сейчас бесплатный доступ к патентной информации на сайте Роспатента можно получить лишь через открытые реестры. Однако поиск по ним возможен только по номеру документа. Очевидно, что в таком режиме даже самое простое патентное исследование невозможно из-за необходимости просмотра огромного количества документов.

Реестрам, предоставляемым на сайте Роспатента, необходимо придать статус официальной публикации, так как без наличия такого статуса они являются по существу демонстрационной базой,

что вынуждает пользователя обращаться к бумажной документации (на сайте существует предупреждение о том, что сведения, размещенные в открытых реестрах, не являются официальной публикацией и могут отличаться от сведений, содержащихся в официальных документах).

С технической точки зрения для удобства информационного поиска и повышения его эффективности целесообразно проведение следующих мероприятий: обеспечение раздельного поиска по базе патентов и заявок на патент; объединение большого количества разрозненных баз (только информация об изобретениях распределена между пятью базами), увеличение количества документов, выдаваемых для просмотра по результатам поиска (в настоящее время их число составляет 200 документов); совершенствование интерфейса поиска и увеличение доступной скорости просмотра документов.

Представляется, что предложенное развитие системы патентной информации станет важным фактором обеспечения инновационного развития страны.

Примечания

[1] См.: Данные по странам Европейского союза по итогам европейского обследования инноваций за период 2002–2004 гг. (источник Евростат) [Электронный ресурс] // Сайт OECD. [М., 2009]. URL: http://www.oecd.org (дата обращения: 4.11.2009).

[2] См.: WIPO Patent report: Statistics on World Wide Patent Activity (2007 Edition) [Электронный ресурс] // Сайт WIPO. [М., 2009]. URL: http://www.wipo.int/ipstats/en/statistics/patents (дата обращения: 4.11.2009).

М.И. Забежайло

К ВОПРОСУ О ВЫБОРЕ АДЕКВАТНОЙ МЕТОДОЛОГИИ ТРАНСФОРМАЦИИ БИЗНЕСА КРУПНОГО КОММЕРЧЕСКОГО БАНКА

Обсуждаются особенности процесса модернизации ИТ-инфраструктуры российских коммерческих банков. Выделен набор критически важных для успеха подобных проектов требований. Предложены варианты минимизации наиболее существенных архитектурных и организационных проектных рисков.

Ключевые слова: ИТ-инфраструктура банковского бизнеса, модернизация ИТ-инфраструктуры, архитектура банковских информационных систем, компонентно-интеграционный подход, контроль и минимизация проектных рисков.

Одним из очевидных следствий текущего экономического кризиса оказалось все более ощутимое формирование новой экономической реальности, важнейшими особенностями которой стали внимание к прозрачности организации и эффективности управления бизнесом, контролю издержек, оптимизации инфраструктуры и технологий поддержки бизнеса.

В финансовой сфере эту складывающуюся на наших глазах новую экономическую реальность характеризуют в том числе тенденции укрупнения банковских структур, слияний и поглощений, увеличения требований к капиталу и т. п. Кризис запустил дополнительные механизмы консолидации банковской системы. В этой ситуации в области информационных технологий у новых владельцев бизнеса обычно формируется достаточно серьезный комплекс задач, связанных с интеграцией в единое целое нескольких разнородных ИТ-инфраструктур, а также множества наследуемых в процессе консолидации бизнес-процессов и прикладных систем[1].

В части конкретных бизнес-процессов одним из первых в фокус внимания попадает, например, комплекс вопросов, связанных с кредитованием реального сектора экономики. Для многих банков сегодня успешная деятельность на этом направлении требует существенного изменения подходов и технологий управления рисками. Резкое возрастание кредитных рисков стало важнейшей отличительной особенностью того особого положения, которое занимают сегодня такие банки[2]. В этих условиях совершенствование технологий и расширение продуктового ряда вместе с развитием обеспечивающей их ИТ-инфраструктуры – эффективный путь, позволяющий обеспечить устойчивое развитие бизнеса банка на длительную перспективу. В свою очередь, наблюдаемый сегодня на ИТ-рынке всплеск интереса к решениям в области банковской аналитики и управления рисками отражает осознанный уровень понимания важности этой проблемы и, как следствие, растущий объем инвестиций крупных банков в решения названного типа.

В складывающихся условиях одним из приоритетов государства на ближайшую перспективу обозначено «повышение устойчивости национальных финансовых институтов, формирование мощной финансовой системы как надежной основы для развития национальной экономики». Принятая руководством страны Программа антикризисных мер, в частности, провозглашает, что «Правительство Российской Федерации и Банк России будут стимулировать консолидацию в банковской сфере, формирование крупных и финансово устойчивых банковских структур, конкурентоспособных на международном уровне...». Здесь, помимо прочего, «предусматривается возможность выделения в 2009 году 495 млрд рублей на поддержку банковской системы, в том числе 280 млрд рублей – на капитализацию банков, 215 млрд рублей – фондирование за счет Фонда национального благосостояния. При этом предоставление государственной поддержки будет увязано с кредитованием реального сектора...». А в части кредитования и поддержки национальных сельхозпроизводителей отмечается, что «...дополнительно капитализированы открытое акционерное общество “Россельхозбанк” (на 45 млрд рублей) и открытое акционерное общество “Росагролизинг“ (на 25 млрд рублей)...»[3]

Итак, на повестке дня – изменения требований к ИТ-инфраструктуре российских финансовых институтов. Кризис изменил расстановку приоритетов, выявил новые (по отношению к докризисным временам) потребности и в организации банковских бизнес-процессов, и в информационных технологиях. Сегодня перед ведущими российскими банками во все более явной форме стоит задача выведения их бизнеса, а также ИТ-инфраструктуры на уро-

вень сложности и функциональности, соответствующий потребностям и текущей практике деятельности лучших финансовых институтов развитых стран мира.

Таким образом, впереди серьезные преобразования технической и бизнес-инфраструктуры ведущих российских банков. Как, выстраивая процесс модернизации в конкретной банковской организации, сформировать нужный функционал целевой ИТ-системы? Как надежно контролировать при этом бюджет и сроки исполнения модернизационного проекта? Активный поиск приемлемых ответов на эти вопросы сегодня – один из основных приоритетов российского рынка банковских информационных технологий.

Специфика обсуждаемой модернизации как технологического и организационного процессов в первую очередь – в необходимости «оперировать пациента» прямо в процессе его производственной деятельности. Действительно, трудно представить себе ситуацию, когда кто-либо из лидеров рынка банковских услуг мог бы позволить себе покинуть его хотя бы на некоторое время (для проведения необходимого реинжиниринга собственной инфраструктуры), чтобы затем вернуться в этот же бизнес со вновь приобретенными технологическими возможностями. На практике приходится вести модернизационные преобразования, одновременно поддерживая наследуемые (из текущего ИТ-ландшафта) программные компоненты и разрабатывая новые, при этом обеспечивая эффективную интеграцию программных приложений обоих названных типов. Разумеется, весь спектр традиционных факторов риска, свойственных подобной деятельности, – от области *поддержания непрерывности бизнеса*[4] до, например, обеспечения *информационной безопасности* и *защиты данных* для вновь разработанных прикладных программных компонентов или же необходимости поддерживать *целостность* формируемого целевого ИТ-комплекса и других – остается актуальным для исполнителей обсуждаемых проектов модернизации банковского бизнеса.

Как обеспечить согласованное функционирование различных программных компонентов? Как наладить эффективное взаимодействие наследуемых и вновь разработанных информационных систем, сводимых в единый комплекс в процессе слияний и укрупнений? Как найти программные компоненты для целевого решения, которые могли бы сочетать в себе и характеристики стандартизованного индустриального продукта, и достаточно высокий уровень конкретной проблемной ориентированности? Наконец, как подобрать оптимальную для конкретного банка последовательность ввода в промышленную эксплуатацию прикладных подсистем – составных частей создаваемого целевого решения? Эти и

многие другие аналогичные вопросы должны быть снабжены адекватными ответами (желательно!) еще до начала соответствующего модернизационного проекта.

Рассматриваемые нами группы вопросов, разумеется, отражают различные аспекты единого процесса – комплекса модернизационных преобразований. Таким образом, их взаимная зависимость, а вместе с нею и взаимозависимость ответов на эти вопросы – естественная особенность обсуждаемого процесса модернизации. В такой ситуации представляется целесообразным выстраивать *единый комплексный подход* к организации рассматриваемых модернизационных преобразований. В основании такого подхода естественно поместить соответствующую методологию трансформационных преобразований.

Какой ей следует быть (каковы ее детальная структура и необходимые подробности описания)? Как обеспечить *результативность* такой методологии (другими словами, как обеспечить, чтобы четкое следование рекомендациям такой методологии гарантировало достижение *позитивного результата* в соответствующем процессе модернизации)? Чем собственно отличается подобная методология от всего остального – принципов, приемов и методологий управления проектами, накопленных ИТ-индустрией за десятилетия ее существования? Наконец, каким (желательно, хорошо формализованным) *требованиям* должна удовлетворять подобная результативная методология?

Предлагаемый здесь вариант ответа на эти вопросы основывается на следующей идее: подход к формированию требований, характеризующих результативную методологию модернизационных преобразований рассматриваемого нами типа, должен учитывать *все наиболее существенные риски* процесса модернизации. Эффективно управляя проектными рисками модернизации, располагая надежными механизмами и средствами компенсации выявленных рисков, можно достичь желаемых результатов – гарантированно привести соответствующие трансформационные преобразования к намеченным целям. Итак, попробуем проследить цепочки взаимосвязей от (выявленных в значительной мере эмпирическим путем[5]) наиболее существенных проектных рисков к механизмам и средствам их (рисков) компенсации, а затем от механизмов управления выявленными рисками модернизации к требованиям, характеризующим обсуждаемую *результативную* методологию.

Продвигаясь вперед, к только что обозначенным целям, обратимся к обзору (по крайней мере, необходимой части) карты проектных рисков, требующих внимательного учета, анализа и адекватного управления в ситуации, когда наша цель – успех проекта

модернизации бизнес- и ИТ-инфраструктур крупного коммерческого банка. В предлагаемом анализе мы сначала будем рассматривать своего рода «операционный контекст» возникновения соответствующих рисков и лишь затем – явный вид связанных с таким контекстом угроз, недостаточное внимание к которым может поставить под сомнение успех требуемых модернизационных преобразований.

Процессная модель бизнеса: проблема интеграции процессов при осуществлении модернизационных преобразований

Опыт модернизационных проектов новейшего времени как в российских, так и в зарубежных банках показывает, что критически важным направлением осуществляемых здесь преобразований оказывается организация взаимодействия различных бизнес-процессов, соединяемых во взаимносогласованный комплекс в рамках целевой бизнес-, а также (обеспечивающей исполнение соответствующих бизнес-процедур) ИТ-инфраструктуры. Выстраивание единой целостной карты бизнес-процессов, согласование (в рамках единой целевой информационной системы) используемых в каждом из них бизнес-логики и моделей данных – вот весьма чувствительная область организации обсуждаемых модернизационных преобразований. Без отлаженности процессной инфраструктуры бизнеса трудно рассчитывать на его операционную эффективность[6]. Без унификации моделей данных, используемых различными программными приложениями, трудно рассчитывать на согласованную работу программных компонентов целевой ИТ-системы. Именно по этим причинам принципиально важно еще на стадии проектирования ИТ-архитектуры целевого решения уделить особое внимание формальной модели (а в ряде случаев – системе взаимодействующих[7] моделей) данных, «настройка» которой обеспечит эффективную интегрируемость программных компонентов, а вместе с нею и целостность, гибкость (в части настроек «траекторий» тех или иных конкретных бизнес-процессов), а также удобство модернизируемости целевой бизнес- и ИТ-инфраструктур в соответствии с изменяющимися потребностями рынка. На основе подобной формальной модели данных могут быть построены формализованные[8] описания реализуемых бизнес-процессов. В свою очередь, подобные описания – лучшее «руководство» для настройки исполнения целевых бизнес-процессов, а также составляющих их конкретных бизнес-процедур средствами выбранного прикладного программного обеспечения.

Здесь также полезно учитывать, что с формальной точки зрения предлагаемые лидерами рынка объектно-ориентированные прикладные инструментарии представления и моделирования данных[9] имеют сопоставимые выразительные возможности, что позволяет обеспечить их корректное[10] взаимодействие (через соответствующие адапторы) по крайней мере на наиболее существенных фрагментах анализируемой предметной области[11] – построения программных инструментов автоматизации банковского бизнеса.

Проблема подбора необходимых SW-компонентов для полнофункционального целевого решения

Весьма существенную роль в продвижении к успеху модернизационных преобразований имеет выбор прикладной программной платформы для целевой ИТ-системы. Одним из первых требует ясного ответа вопрос о принципе формирования целевого ландшафта ИТ-компонентов: будет ли этот ландшафт *гомогенным* (т. е. построенным на программных решениях одного производителя) или же ему предстоит быть *гетерогенным* – интегрирующим решения различных поставщиков? Параллельно требует внимания проблема правильного подбора необходимых прикладных программных компонентов для полнофункционального целевого решения. Какими средствами обеспечить его целостность (как единого согласованного программного комплекса)?

В самом общем виде путей надежного управления возникающими здесь рисками всего два, причем оба базируются на одной технологии – компонентно-интеграционных моделях сервисно-ориентированной архитектуры[12]. Сегодня даже крупнейшие компании – лидеры ИТ-рынка, предлагая так называемые программные платформы для создания многофункциональных прикладных клиентских решений, не рискуют утверждать, что их продуктов достаточно для удовлетворения потребностей любого клиента. На рынке существует множественность предложений, когда одни и те же элементы «карты функциональных потребностей» потенциальных заказчиков могут быть «покрыты» предложениями (индустриальными программными продуктами) различных производителей. Спектр возможных различий (а с ними – и оснований для выбора) оказывается достаточно широк: от объема встроенных функций (характеризующих так называемые «легкие» или же, наоборот, «тяжелые» решения) до соответствующих функциональным возможностям ценовых показателей, от ориентированности на гибкость инструментальных возможностей до встраивания в программные

системы конкретных версий соответствующих прикладных бизнес-процессов[13] и т. п. Тем не менее, главным в принятии решения о выборе программных компонентов целевого решения должно, по-видимому, быть внимание к SOA[14] – характеристикам составных частей формируемой прикладной информационной системы, а также лежащим в их основе формализованным моделям данных, к конкретным техническим особенностям ESB[15]-интеграции, к возможностям удобного участия в процедурах так называемой «оркестровки» (то есть гибкой настройки специальными ESB-средствами) реализуемых целевой архитектурой прикладных бизнес-процессов.

Дополнительные преимущества при реализации подобного подхода удается получить, если стартовать с использования индустриальной программной платформы какого-либо из лидеров ИТ-рынка[16], так сказать, «до-интегрируя» (типовыми ESB-средствами) к базовому ИТ-ландшафту необходимые дополнительные проблемно ориентированные программные компоненты и обеспечивая подобным образом функциональную «полноту» целевого бизнес-решения. (Использование здесь индустриальной программной платформы во взаимодействии с индустриальным интеграционным инструментарием позволяет обеспечить высокий уровень надежности целевого ИТ-комплекса в условиях, когда сроки его реального формирования, благодаря использованию типовых индустриальных решений, оказываются также надежно контролируемыми.)

Проблема проектирования целевых бизнес-процессов

Одним из традиционно трудных аспектов в практике крупных проектов разработки и внедрения ИТ-систем является проблематика формирования соответствующей карты бизнес-процессов, реализовывать которые и призвана целевая информационная система. Ставшие уже «классическими» трудные места здесь – это, помимо прочего:

- размеры и распределенная «топология» объекта автоматизации, следствием которой достаточно часто оказываются требования учета той или иной локальной специфики ведения бизнеса (а вместе с ними и требования внести те или иные изменения в «типовую» схему бизнес-процессов, отказаться от их унификации и т. п.);
- проблемы с согласованием и утверждением (в рамках крупной территориально распределенной структуры объекта авто-

матизации) единой карты целевых бизнес-процессов (как соответствующего документа НСИ организации);
– проблема качества (операционной эффективности) предлагаемых к реализации целевых бизнес-процессов.

Сегодняшний уровень развития ИТ-индустрии дает возможность принимать в такой ситуации достаточно эффективные решения, базирующиеся на использовании программных инструментов нового поколения. Компании – лидеры индустрии уделяют особое внимание обобщению накопленного ими опыта успешных проектов обсуждаемого типа. Основное направление развития технологий здесь выглядит следующим образом: с учетом накопленного практического опыта формируются проблемно ориентированные типологии бизнес-процедур вместе с лежащими в их основе моделями и типологиями данных (в частности, характерных для конкретной предметной области объектов, атрибутов, отношений, процессов и т. п.). Порождаемые таким путем типологии дополняются средствами графического представления и (если потребуется) модификации моделируемых бизнес-процедур. В результате этих действий порождается понятийная и технологическая «платформа» для формирования так называемых репозиториев типовых бизнес-процессов и процедур, в которых отражаются проверенные временем лучшие практики (так называемые процессные best practices) компании-разработчика, накапливаемые в соответствующей области проблемно ориентированных приложений.

Далее (в рамках SOA-технологий) каждому из типовых фрагментов рассматриваемых бизнес-процессов сопоставляется соответствующий программный компонент (web-сервис, реализующий алгоритмику сопровождаемого им бизнес-компонента). Таким образом, процесс визуального проектирования бизнес-процесса (средствами соответствующего программного инструментария[17]) из «типовых фрагментов» – элементов накапливаемого проблемно ориентированного рипозитория бизнес-процедур – сопровождается процессом подбора соответствующих программных компонентов и затем порождения исполнимого кода соответствующего фрагмента целевой информационной системы. Таким образом, на базе SOA-технологий в рамках компонентно-интеграционной архитектуры целевое решение формируется путем «программирования без программирования»[18]. При этом представленной технологии формирования целевой программной системы оказывается свойственен ряд весьма привлекательных особенностей, в частности:
– в основе целевого решения – проверенные практикой фрагменты исполнимого кода, реализующего операционно эффективные бизнес-действия;

– для формирования исполнимого кода целевого решения не требуется детального знания особенностей тех или иных разделов технологии программирования[19];
– если возникает необходимость внесения тех или иных изменений в исполнимый код соответствующего программного приложения[20], то используемый подход гарантирует гибкость и простоту решения возникающей задачи. Действительно, все требуемые процедуры могут быть легко реализованы той же технологией «программирования без программирования»: визуальное перепроектирование соответствующего процесса (например, компоновка некоторых его частей из других «типовых блоков»), затем ESB-«сборка» вновь отобранных программных компонентов – и обновленное решение готово.

Здесь следует особо подчеркнуть ключевую роль уже обсуждавшейся выше единой (сквозной) модели данных для успеха интеграции программных компонентов целевой прикладной ИТ-системы. Для собранных в репозитории типовых бизнес-процессов именно единая модель данных оказывается процедурным основанием эффективной интеграции соответствующих прикладных программных компонентов. Наличие такой модели – ключевой фактор успеха соответствующего модернизационного проекта.

Не менее существенны функциональные свойства обсуждаемой компонентно-интеграционной архитектуры с точки зрения предоставляемых ею возможностей гибкой настройки новых версий бизнес-процессов. В рассматриваемой технологии интеграционная шина предприятия превращается в своего рода бизнес-хаб, на уровне которого текущая «оркестровка» – та или иная оперативная перенастройка – корпоративных бизнес-процессов становится гибкой, компактной по времени и достаточно надежно управляемой бизнес-процедурой[21].

Проблема наследования эффективных программных компонентов действующего ИТ-ландшафта в целевую ИТ-систему

Опыт текущих модернизационных проектов в крупных российских банках показал, что одним из наиболее чувствительных аспектов осуществляемых трансформационных преобразований оказалась проблема наследования (при переходе от текущего ИТ-ландшафта к целевому) тех программных компонентов, демонстрируемая в ежедневной работе эффективность которых позволяет использовать их по крайней мере в промежуточных, а в ряде случаев – и в целевых архитектурных решениях. Необходимость

учитывать в процессе модернизации такие очевидные требования, как, например, жесткий контроль и экономия расходов, или же такие принципы, как, например, «модернизируя ИТ-компоненты, не навреди основному бизнесу в его текущем состоянии», – вот очевидные примеры аргументов в пользу особого внимания к этой проблематике.

В отличие от традиционных для российской банковской отрасли «монолитных» решений SOA-технологии предоставляют ряд интересных возможностей для надежного контроля и управления проектными рисками. Действительно, компонентизация приложений (понимаемая как разделение бизнес-функций действующей ИТ-системы на взаимосвязанные группы, внутренние связи в которых более существенны, чем связи между выделяемыми группами[22]), выстраивание их взаимодействий через стандартизованную ESB-среду, а также разделение выделенных компонентов на те, которые требуют оперативной замены (модернизации), и те, которые могут быть «наследованы» в следующих версиях архитектуры ИТ-систем предприятия, – вот принципиальная схема решения сформулированной выше проблемы наследования эффективных ИТ-компонентов.

Заметим, что предложенная последовательность действий позволяет достаточно надежно контролировать риски обсуждаемых модернизационных трансформаций текущего ИТ-ландшафта, учитывая при этом и сформулированные выше ограничения, и операционные принципы управления планируемыми изменениями.

Не менее чувствительная область проектных рисков – проблематика совместимости компонентов целевой ИТ-системы при ее «сборке» из наследуемых и вновь разработанных программных модулей. Архитектурная целостность целевого решения[23] обычно устанавливается в процессе интеграционных тестов. При последовательной разработке «монолитного» архитектурного решения необходимость регулярного проведения таких тестов – один из ключевых элементов управления рисками проекта. (Так, например, известны случаи, когда отказ от тех или иных промежуточных интеграционных тестов в ситуации нештатного развития событий требовал существенного возврата назад в проектных планах, так как вовремя не выявленные погрешности в организации взаимодействия создаваемых программных компонентов фатальным образом влияли на все последующие проектные разработки.)

При использовании основанного на SOA подхода эта проблематика находит совершенно иное (причем достаточно элегантное) решение. Здесь фактически успех интеграционных тестов

является техническим следствием правильной организации трех факторов:

– корректного согласования моделей данных интегрируемых компонентов;
– корректного (исключающего ошибки алгоритмики взаимного преобразования данных) формирования соответствующих адапторов, обеспечивающих взаимодействие каждого из интегрируемых программных компонентов с интеграционной шиной предприятия;
– корректной настройки (так называемой «оркестровки») соответствующих бизнес-процессов с помощью корпоративного бизнес-хаба – интеграционной шины предприятия.

При соблюдении этих трех условий имеется возможность последовательного присоединения новых программных компонентов к уже сформированному (компонентно-интеграционному!) ИТ-ландшафту. В этих условиях скорость информационного обмена между компонентами (вместе с показателями производительности всего полученного в результате покомпонентной «сборки» целевого программного комплекса) будет определяться производительностью корпоративной интеграционной шины[24] (а это уже достаточно хорошо разработанная область применения типовых индустриальных технологий и решений).

Выбор стратегии комплексной модернизация ИТ-ландшафта: все сразу или по частям?

Одним из первых в «запускаемом в работу» проекте модернизации приходится решать вопрос о выборе последовательности основных шагов трансформационных преобразований: что, за чем, в какой последовательности и т. п. В самом общем виде – это проблема выбора: модернизировать *все сразу* или же *по частям*? В более аккуратной постановке – это вопрос о порядке разработки, тестирования и вывода из тестового режима эксплуатации в продуктивный тех или иных программных компонентов целевой ИТ-системы. К сожалению, в подавляющем большинстве проектов перспектива построить рядом с эксплуатируемой системой новую, оттестировать ее, обучить эксплуатационный персонал и одним щелчком рубильника переключиться со старой ИТ-инфраструктуры на новую оказывается всего лишь романтической иллюзией. Реальная бизнес-практика, как правило, требует «оперировать пациента» прямо в ходе его производственной деятельности. В этих условиях выбор архитектурного решения для целевой ИТ-системы

(например, «монолит» или же конфигурация на базе SOA-подхода) оказывается критически важным моментом для всей последующей проектной работы. Этим выбором также определяются и основные типы проектных рисков, и важнейшие механизмы противодействия их реализации.

Оставляя за рамками обсуждения альтернативные архитектурные решения, обратимся к укрупненному обзору технологических и организационных преимуществ компонентно-интеграционного (основанного на SOA) подхода.

Прежде всего отметим уже упоминавшиеся выше возможности упрощения интеграционных тестов – проверок целостности собираемой многокомпонентной целевой ИТ-системы. Отдельно подчеркнем возможности «доинтеграции» необходимых программных компонентов к уже имеющемуся ИТ-ландшафту, что позволяет минимизировать риски внесения тех или иных изменений в наследуемые части эксплуатируемого программного комплекса.

Не менее важной организационной особенностью такой стратегии исполнения обсуждаемых проектов трансформационных преобразований оказываются возможности *покомпонентной* модернизации уже эксплуатируемой ИТ-системы банка. Компонентно-интеграционный ландшафт (архитектурно-технологическая платформа вида CBM[25] + ESB) уже зарекомендовал себя как надежная основа для организации «пошагового» процесса реализации требуемых модернизационных трансформаций. Важнейшей отличительной характеристикой такого подхода оказываются возможности *разделять* (локализуя их в рамках соответствующих «шагов» процесса модернизации[26]) и надежно *контролировать* проектные риски. Таким путем, в частности, формируются дополнительные инструменты надежного контроля соблюдения установленных сроков и бюджета исполнения проекта. Действительно, технологическая платформа CBM + ESB позволяет провести детальную декомпозицию проектных работ на (легко регулируемые по размеру[27]) функциональные блоки, порядок взаимодействия которых определяется стандартизованными возможностями корпоративной интеграционной шины. В такой ситуации опытная команда проектных менеджеров (следуя, например, известной стратегии «разделяй и властвуй» в идентификации и структуризации проектных рисков) имеет все необходимые основания продемонстрировать мощь и преимущества современных PM-технологий[28] в организации надежного продвижения соответствующего модернизационного проекта к запланированным целям.

Проблема доработки исходного кода базового программного обеспечения

Опыт российских модернизационных проектов последнего времени[29] выявил критическую значимость проблемы доработки исходного кода того или иного прикладного программного обеспечения, признанного базовым в соответствующем проекте преобразования банковского бизнеса[30]. Потребность в подобных доработках связана, в первую очередь, с необходимостью привести предназначенное для использования в режиме продуктива банковское ПО в соответствие с требованиями российских государственных регуляторов. Важнейшие области доработки здесь – это соответствие требованиям российского бухгалтерского и налогового учетов, а также проблематика обеспечения информационной безопасности.

В «операционно-процедурном» плане возникающие здесь риски обусловлены следующими двумя обстоятельствами:

- в соответствии с действующим законодательством в области авторского права внесение тех или иных изменений в программное обеспечение возможно лишь с разрешения правообладателя;
- появление в исходном коде соответствующего программного обеспечения тех или иных изменений, не санкционированных правообладателем, позволяет последнему отказаться от стандартных (и обязательных для него!) услуг по сопровождению того программного продукта, который он поставляет на рынок.

Фактически в такой ситуации исполнители модернизационного проекта оказываются в «тисках» достаточно жестких ограничений: договаривайся с правообладателем о внесении изменений в исходный код прикладного ПО (а это может оказаться организационно сложной и коммерчески дорогостоящей процедурой), либо готовься все эксплуатационные риски использования измененного ПО нести на себе[31].

Компонентно-интеграционный подход предоставляет достаточно элегантные возможности минимизации рисков только что описанного типа. Технология компонентизации, о которой мы уже говорили выше, позволяет группировать бизнес-функции соответствующих приложений таким образом, чтобы исполняющие их комплексы web-компонентов можно было компоновать из уже имеющихся и вновь созданных программных модулей. В такой ситуации, сохраняя уже имеющиеся в базовом программном обеспечении web-сервисы, достаточно дополнить их вновь созданными сервисами, которые корректно взаимодействуют через ин-

теграционную шину с уже имеющимися программными компонентами. Таким способом можно формировать функционально-корректные ландшафты программных приложений, не затрагивая при этом защищенной законодательством об авторских правах интеллектуальной собственности компании-вендора, производящего используемое в соответствующем модернизационном проекте базовое ПО.

В части работ по обеспечению информационной безопасности компонентно-интеграционные архитектуры приложений предоставляют дополнительные возможности за счет использования интеграционной шины предприятия как платформы для размещения и интеграции соответствующих ИБ-сервисов[32]. При этом при выстраивании соответствующих систем защиты не требуется «глубокого внедрения» в каждый элемент структуры базового прикладного ПО. Во многих случаях здесь достаточно лишь контролировать «оркестрованный» через интеграционную шину информационный обмен между программными компонентами-сервисами, открывая (или же, наоборот, запрещая) доступ тем или иным процессам к тем или иным программным компонентам, а также данным (необходимым для штатной работы приложений информационным ресурсам).

Таким образом, компонентно-интеграционная архитектура приложений в целом ряде случаев позволяет при решении задач обеспечения информационной безопасности уйти от *полного* анализа исходного кода защищаемых программных приложений. Основное внимание здесь фокусируется на соответствующих (функциональности защищаемой информационной системы) моделях угроз и технологиях их нейтрализации. При этом комбинируется частичный анализ исходного кода ПО (например, при анализе интерфейсов взаимодействия интеграционной шины и программных компонентов-сервисов) и проблемно ориентированные технологии риск-менеджмента («эшелонированное» применение которых позволяет минимизировать вероятность преодоления нарушителем создаваемых средств защиты). Функциональная гибкость компонентно-интеграционных архитектурных решений предоставляет дополнительные удобства для модификации или же оперативной замены тех или иных компонентов системы обеспечения информационной безопасности в целевом ИТ-ландшафте. Ее компонентно-модульный характер позволяет поддерживать надежность и целостность защиты путем целенаправленных *локальных* модификаций, обеспечивая оптимизацию показателей цена–качество соответствующего программно-аппаратного ИБ-комплекса.

Заключение

В заключение несколько соображений, связанных со встречающимися в ряде случаев рисками организационного характера. Опыт уже называвшихся выше проектов модернизации показал, что, к сожалению, достаточно распространенной в российских банках проблемой оказывается (явное или неявное) противодействие «полевых командиров» бизнеса предлагаемой руководством модернизации.

Социологические, а также психологические основания подобного противодействия, по-видимому, заслуживают отдельного обсуждения (и еще только ожидают своего заинтересованного исследователя).

В рамках предпринятого нами технического и PM-анализа представляется целесообразным лишь обратить внимание на дополнительные возможности минимизации негативных «социальных» факторов при выборе проектной стратегии *пошаговых изменений* текущих бизнес- и ИТ-ландшафтов предприятия. Очевидно, что технология покомпонентной модернизации позволяет, помимо прочего, надежно контролировать также риски и угрозы организационного противодействия реализуемым трансформационным изменениям. Обсуждавшиеся выше методики декомпозиции проектных преобразований, по-видимому, могут оказаться полезными в управлении и такими – весьма своеобразными – рисками, в основе которых отмеченные выше факторы «социального» характера.

С точки зрения естествоиспытателя уместен вопрос о *достаточности* предлагаемых требований к обсуждаемой результативной методологии трансформационных преобразований. Управление проектами (и в том числе управление проектами модернизации информационных систем коммерческого банка) – сложившаяся область деятельности, объединяющая определенный технический инструментарий (который в ряде фрагментов рассматриваемой предметной области достаточно детально формализован) с *искусством* менеджера. Можно ли в таких условиях надеяться на *достаточность описания* обсуждаемой методологии (в частности, методов и приемов, характеризующих демонстрируемое конкретным менеджером *искусство управлять*)? По-видимому, утвердительно на поставленный вопрос можно ответить лишь в небольшом числе конкретных контекстно-зависимых случаев. Таким образом, предлагаемые выше требования представляется целесообразным[33] считать составной частью *необходимых* для успеха модернизации *условий*: невнимание хотя бы к некоторым из них резко повышает риски соответствующего проекта не прийти к успешному завершению.

1 См.: *Корнильев К.Г.* Банки являются самыми крупными потребителями IT [Электронный ресурс] // Сайт «Интеллектуальный банк». [М., 2009]. URL: http://int-bank.ru/analyst/100/ (дата обращения: 4.11.2009); *Сенаторов М.Ю.* Приоритеты меняются... [Электронный ресурс] // Сайт «Интеллектуальный банк». [М., 2009]. URL: http://int-bank.ru/analyst/94/ (дата обращения: 4.11.2009).

2 Там же.

3 См.: Программа антикризисных мер Правительства Российской Федерации на 2009 год. 19 июня 2009 года [Электронный ресурс] // Сайт Правительства РФ. [М., 2009]. URL: http://www.premier.gov.ru/anticrisis/ (дата обращения: 4.11.2009).

4 Сопровождения и поддержки функционирования уже эксплуатируемых ИТ-систем, обеспечения их операционной надежности в процессе предоставления бизнесу необходимых ИТ-сервисов и т. п.

5 См.: Бинбанк, совместно с партнерами, запустил беспрецедентно многофункциональную банковскую ИТ-систему. Бинбанк: Новости Банка, 20.05.2009. [Электронный ресурс] // Сайт Бинбанка. [М., 2009]. URL: http://www.binbank.ru/rus/news/news_bank/article.wbp?article-id=5BF9896F-937D-4BA9-97F6-9472E23A1737 (дата обращения: 4.11.2009); Бинбанк внедрил многофункциональную банковскую ИТ-систему. CNews, 21.05.2009. [Электронный ресурс] // Сайт журнала CNews. [М., 2009]. URL: http://www.cnews.ru/news/line/ index.shtml?2009/05/21/347961 (дата обращения: 4.11.2009); *Легезо Д.* Иностранное ПО в банках: без россиян никак [Электронный ресурс] // Сайт журнала CNews. [М., 2009]. URL: http://www.cnews.ru/news/top/index.shtml? 2009/05/21/348043 (дата обращения: 4.11.2009); см.: *Павлова О.* ОТП Банк модернизирует технологическую основу розничного бизнеса. [Электронный ресурс] // Сайт журнала PCWeek. [М., 2009]. URL: http://www.pcweek.ru/themes/detail.php?ID=119314&THEME_ID=13884 (дата обращения: 4.11.2009); *Попова М.* Банки «перекраивают» свои ИТ-ландшафты [Электронный ресурс] // Сайт журнала CNews. [М., 2009]. URL: http://www.cnews.ru/ reviews/free/banks2008/articles/recover.shtml (дата обращения: 4.11.2009); *Забежайло М.И.* Банковский бизнес в России: индустрия или искусство? Волгоград: Волгоградское научное издательство, 2009.

6 Именно по этой причине лидеры ИТ-индустрии уделяют специальное внимание обобщению и формализации опыта лучших проектов, формированию специализированных центров компетенции, где накапливаются формализованные модели бизнес-процессов (включая алгоритмические описания и соответствующие программные библиотеки) – все лучшее, что зарекомендовало себя в конкретных проектах. См.: IBM Industry Models for Financial Services. The Information FrameWork (IFW) Overview. Dublin, Ireland: IBM Financial Services Solution Centre, 2004; *Bonati B., Regutzki J., Schroter M.* Enterprise Services Architecture (ESA) for Financial Services: Taking SOA to the Next Level (Paperback). N. Y.: SAP Press & Galileo Press, 2006.

7 Имеется в виду технология интеграции приложений средствами Enterprise Service Bus (Интеграционной шины предприятия), в том числе технология доменной интеграции, когда целевая ИТ-система строится на нескольких доменах вида <ESB + замкнутые на нее программные компоненты>. При этом интегрируемые домены взаимодействуют здесь между собой не только в режиме *точка=>точка*, но и по схеме *много=>во много*. См.: *Chappell D.A.* Enterprise Service Bus. O'Reilly Media, 2004; *Gable J.* Enterprise application integration // Information Management Journal. 2002. March–April; *Bonati B., Regutzki J., Schroter M.* Op. cit.; *Забежайло М.И.* Указ. соч.

8 Например, в виде соответствующих документов нормативно-справочной информации банка – детальных (и алгоритмически корректных) операционных регламентов, правил и т. п., – разработанных с применением современных программных инструментальных средств описания и моделирования бизнес-процессов. (См., в частности, описания продуктов IBM Rational или SAP BPM и др.) См.: IBM Industry Models for Financial Services; *Bonati B., Regutzki J., Schroter M.* Op. cit.

9 См., например, уже упоминавшиеся выше технологии компаний IBM и SAP AG. См.: IBM Industry Models for Financial Services; *Bonati B., Regutzki J., Schroter M.* Op. cit.; SAP for Banking: Media Coverage (2007 Banking Media Highlights). SAP Global Communications / Ed. by T. Balgheim & A. Robillard. 2007; Решение SAP для банковского бизнеса. М.: SAP AG & SAP CIS, 2008. См. также: URL: http://www.sap.ru.

10 То есть такое, при котором объектам (а также атрибутам, отношениям, процессам и т. п.) одной модели данных могут быть сопоставлены соответствующие конструкции в другой модели данных и наоборот.

11 Именно таким путем, в частности, можно строить *доменные* интеграционные структуры (см. сноску выше): здесь программные компоненты, базирующиеся на соответствующей модели данных, интегрируются каждая на «свою» шину, а «согласующие» адапторы обеспечивают взаимодействие программных приложений уже на уровне информационного обмена между задействованными интеграционными шинами («доменами интеграции»).

12 См.: *Bonati B., Regutzki J., Schroter M.* Op. cit.; *Забежайло М.И.* Указ. соч.

13 Как правило, производитель программного обеспечения (ПО) в таких ситуациях специально подчеркивает «чемпионские» характеристики встроенных моделей бизнес-процессов (то есть то, что обычно обозначают термином best practices – лучшие практики).

14 Service Oriented Architecture – сервисно-ориентированная архитектура программных приложений.

15 Enterprise Service Bus – интеграционная шина предприятия.

16 Прекрасный вариант такого решения для банковской и финансовой сфер (в том числе со столь злободневными для российского банковского рынка возможностями программных сервисов параллельного ведения учетов в стандартах IAS/IFRS и РСБУ) предлагает компания SAP AG. См.: SAP for Banking: Media Coverage; *Bonati B., Regutzki J., Schroter M.* Op. cit.

17 См., например, (как и ранее) описание возможностей специализированных инструментариев IBM Rational или SAP BPM.

18 Действительно, ведь исходный код целевой программной системы порождается «сборкой из типовых кусков», то есть без прямого построчного порождения исполнимого кода соответствующих приложений.

19 Например, не требуется специальных знаний, характерных для областей системного или прикладного программирования.

20 Например, в ситуации, когда потребности рынка требуют внесения тех или иных изменений в текущую карту бизнес-процессов объекта автоматизации, и т. п.

21 То есть процедурой, в которой решающую роль играют уже не сотрудники ИТ-службы, а, например, бизнес-технологи (или же кто-то иной из сферы управления собственно бизнесом).

22 Далее подобные связи стремятся стандартизовать, обеспечив таким образом взаимодействие между компонентами (взаимосвязанными группами бизнес-функций) через типовые интерфейсы и специализированные интеграционные среды.

23 И прежде всего – корректность, надежность и эффективность взаимодействия программных компонентов вновь формируемой программной системы.

24 Варианты технологических решений, позволяющих эффективно управлять производительностью интеграционной шины предприятия, можно найти, например, в работах D.A. Chappell, J. Gable, N. Bieberstein, B. Bonati и других.

25 Component Based Model – компонентная модель.

26 Прежде всего как а) риски корректного формирования соответствующего программного компонента плюс б) риски организации корректного взаимодействия этого компонента с интеграционной шиной предприятия.

27 Фактически размер каждого такого программного модуля определяется (и регулируется) объемом тех бизнес-функций, которые предстоит реализовать средствами соответствующего программного web-сервиса.

28 Project Management – управление проектами.

29 См.: Бинбанк, совместно с партнерами...; См.: Бинбанк внедрил многофункциональную... *Легезо Д.* Указ. соч.; *Павлова О.* Указ. соч.; *Попова М.* Указ. соч.; *Забежайло М.И.* Указ. соч.

30 В первую очередь эта проблема актуальна для программных продуктов западных производителей и обусловлена их ориентированностью на международные стандарты внутреннего и бухгалтерского учетов IAS/IFRS. В свою очередь в части обеспечения информационной безопасности обсуждаемая проблематика, пожалуй, в не меньшей мере актуальна для программной продукции и международных, и российских производителей.

31 А помимо утраты сопровождения здесь в ряде случаев проявляются, например, риски несовместимости измененной (на конкретном проекте) версии соответствующего ПО с новыми стандартными версиями поставляемого вендором базового ПО и т. п.

32 Информационная безопасность.

33 И опыт ряда конкретных модернизационных проектов дает достаточно веские аргументы в пользу такой точки зрения.

Т.А. Асмолов

ТИПОВЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ УЧРЕЖДЕНИЙ КУЛЬТУРЫ И АНАЛИЗ МЕТОДОВ ИХ ЗАЩИТЫ

В статье рассматриваются типовые информационные системы, применяемые в учреждениях культуры. Анализируются системы защиты рассмотренных систем и принципы, в соответствии с которыми они строятся. Анализ возможных угроз и анализ рисков помогают выбору мер безопасности, которые должны быть осуществлены, чтобы уменьшить риск до приемлемого уровня. Автором проведен анализ возможных угроз информационным системам и комплекс мероприятий, направленных на предотвращение их возникновения. Предложена классификация атак на информационные системы и способов их совершения. В заключение выделены общие стратегии, связанные с принятием решения в условиях риска воздействия на информационные системы.

Ключевые слова: информационные системы, методы защиты информации, угрозы информационной безопасности, методы защиты информационных систем, учреждения культуры, информационные ресурсы, уязвимость информационной системы.

Сегодня специалисты музеев и библиотек так или иначе вынуждены заниматься вопросами обеспечения информационной безопасности. Это обусловлено тем, что в ближайшие сто лет нам придется жить в обществе (среде) информационных технологий, куда перейдут все социальные проблемы человечества, в том числе и вопросы безопасности.

Широкое внедрение информационных технологий в жизнь современного общества, не только в сферу учреждений культуры, привело к появлению ряда общих проблем информационной безопасности. Необходимо:

– гарантировать непрерывность и корректность функционирования важнейших информационных систем (далее ИС), обеспечивающих безопасность людей и экологической обстановки;
– обеспечить защиту имущественных прав граждан, предприятий и государства в соответствии с требованиями гражданского, административного и хозяйственного права (включая защиту секретов и интеллектуальной собственности);
– защитить гражданские права и свободы, гарантированные действующим законодательством (включая право на доступ к информации).

Потенциальная уязвимость ИС по отношению к случайным и предумышленным отрицательным воздействиям выдвинула проблемы информационной безопасности в разряд важнейших, стратегических, определяющих принципиальную возможность и эффективность применения ряда ИС в гражданских и военных отраслях.

Требования по обеспечению безопасности в различных ИС могут существенно отличаться, однако они всегда направлены на достижение трех основных свойств[1]:

– целостности – информация, на основе которой принимаются решения, должна быть достоверной и точной, защищенной от возможных непреднамеренных и злоумышленных искажений;
– доступности (готовности) – информация и соответствующие автоматизированные службы должны быть доступны, готовы к работе всегда, когда в них возникает необходимость;
– конфиденциальности – засекреченная информация должна быть доступна только тому, кому она предназначена.

Для решения проблем информационной безопасности необходимо сочетание законодательных, организационных, технологических и стандартизационных мероприятий.

Информационные системы можно рассмотреть как организационно-технические системы, представляющие собой совокупность следующих компонентов: технические средства обработки и передачи данных; системное и прикладное программное обеспечение (далее ПО); информация на различных носителях; персонал и пользователи системы.

Типовыми структурами таких информационных систем считаются:

– автономные рабочие станции;
– локальные системы коллективного пользования;
– глобальные системы коллективного пользования.

Автономные рабочие станции представляют собой один или несколько персональных компьютеров и т. д., не связанных между со-

бой. На любом из них пользователи работают раздельно во времени[2]. Обмен информацией осуществляется через сменные носители. Объектами защиты в автономных рабочих станциях являются: рабочие станции; сменные носители информации; пользователи и обслуживающий персонал; устройства визуального представления информации.

Локальные системы коллективного пользования создаются для коллективной обработки информации и/или совместного использования ресурсов. Оборудование размещено в одном помещении, здании или группе близко расположенных зданий. Структуры локальных систем коллективного пользования:

– без выделенного сервера (одноранговые сети); не требуют централизованного управления. Любой пользователь сам делает свои ресурсы доступными для других. Используется однотипная операционная система (далее ОС);
– с выделенным сервером/серверами. На рабочих станциях и серверах могут быть установлены рабочие станции; требуют централизованного административного управления;
– многотерминальные системы на базе малых и больших компьютеров. Основные ресурсы сосредоточены на сервере. Рабочие станции – терминалы. Общее руководство осуществляет администратор. На центральном компьютере и рабочих станциях используются различные ОС;
– многосегментные локальные сети. Состоят из нескольких сегментов, любой из которых является сетью с выделенным сервером. Объединение осуществляется через мост, в качестве которого может использоваться либо выделенный сервер, либо специальное устройство. Любым сегментом управляет свой администратор. В любом сегменте может использоваться своя ОС;
– смешанные сети; включают все ранее рассмотренные системы.

Объектами защиты в локальных системах коллективного пользования являются:

– все рабочие станции;
– выделенные серверы и центральный компьютер;
– локальные каналы связи;
– реквизиты доступа.

В *глобальных системах* коллективного пользования осуществляется совместная обработка информации и совместное использование ресурсов. Они отличаются от локальных систем тем, что:

– могут находиться на значительном удалении друг от друга;
– каналы связи не принадлежат собственнику системы;
– каналы связи являются коммутируемыми и взаимосвязанными;

– для использования каналов связи необходимо устройство сопряжения;
– подобные системы открытые, и подключиться к ним могут все желающие.

Объектами защиты в глобальных системах коллективного пользования служат те же, что и в локальных системах коллективного пользования (глобальные каналы связи; информация, передаваемая по глобальным каналам связи; информация о реквизитах доступа в глобальные системы коллективного пользования).

Системы защиты для перечисленных объектов строятся в соответствии со следующими принципами[3].

Принцип системности. Системный подход предполагает необходимость учета всех взаимосвязанных взаимодействий и изменяющихся во времени элементов, условий и факторов, существенных для понимания и решения проблемы обеспечения безопасности.

Принцип комплексности. Предполагает строить систему из разнородных средств, перекрывающих все существующие каналы реализации угрозы безопасности и не содержащих слабых мест на стыке отдельных компонентов.

Принцип непрерывной защиты. Защита должна существовать без разрывов в пространстве и времени. Это непрерывный целенаправленный процесс, предполагающий не только защиту в эксплуатации, но и проектирование защиты на стадии планирования системы.

Принцип разумной достаточности. Вложение средств в системы защиты должно быть построено таким образом, чтобы получить максимальную отдачу.

Принцип гибкости управления и применения. При проектировании системы защита может получиться либо избыточной, либо недостаточной.

Принцип открытости алгоритмов и механизмов защиты. Знание алгоритма и механизма защиты не позволяет осуществить взлом системы, в том числе и автору.

Принцип простоты применения защитных мер и средств. Все механизмы защиты должны быть интуитивно понятны и просты в использовании. Пользователь должен быть освобожден от выполнения малопонятной, объемной рутинной работы, так как он не должен обладать специальными знаниями.

Информация в системе, поддержанная информационной технологией, является критическим ресурсом, который позволяет использующим его организациям выполнять свои функции. При этом система будет выполнять эти функции эффективно только при осуществлении надлежащего контроля за информацией, чтобы гаран-

тировать, что она защищена от опасностей нежелательного или несанкционированного распространения, изменения или потери. Мероприятия по обеспечению безопасности предназначены для того, чтобы предотвратить или уменьшить данные и подобные угрозы. Необходимо решать задачи управления распределением средств защиты в организациях культуры, а также систематически проводить анализ возможных угроз и рисков, что, в свою очередь, поможет выбору мер безопасности, которые должны быть осуществлены, чтобы уменьшить риск до приемлемого уровня.

Угроза безопасности компьютерной системы – это потенциально возможное происшествие, неважно, преднамеренное или нет, которое может оказать нежелательное воздействие на саму систему, а также на телеметрическую информацию, хранящуюся в ней.

Уязвимость компьютерной системы – это некая ее неудачная характеристика, которая делает возможным возникновение угрозы. Другими словами, именно из-за наличия уязвимостей в системе происходят нежелательные события.

Наконец, атака на компьютерную систему – это действие, предпринимаемое злоумышленником, которое заключается в поиске и использовании той или иной уязвимости. Таким образом, атака – это реализация угрозы.

Исследователи обычно выделяют три основных вида угроз безопасности: угрозы раскрытия, целостности и отказа в обслуживании.

Угроза раскрытия заключается том, что информация становится известной тому, кому не следовало бы ее знать. Угроза раскрытия имеет место всякий раз, когда получен доступ к некоторой конфиденциальной информации, хранящейся в вычислительной системе или передаваемой от одной системы к другой. Иногда вместо слова «раскрытие» используются термины «кража» или «утечка».

Угроза целостности включает в себя любое умышленное изменение (модификацию или даже удаление) данных, хранящихся в вычислительной системе или передаваемых из одной системы в другую. Обычно считается, что угрозе раскрытия подвержены в большей степени государственные структуры, а угрозе целостности – деловые или коммерческие.

Угроза отказа в обслуживании возникает всякий раз, когда в результате определенных действий блокируется доступ к некоторому ресурсу вычислительной системы. Реально блокирование может быть постоянным, так чтобы запрашиваемый ресурс никогда не был получен, или оно может вызвать только задержку запрашиваемого ресурса, достаточно долгую для того, чтобы он стал бесполезным. В таких случаях говорят, что ресурс исчерпан.

Стоит особо отметить, что уязвимость компьютерной системы делает возможным возникновение угрозы. Таким образом, возникает вероятностная связь между уязвимостью и атакой как реализацией угрозы.

Существуют две классификации атак на ИС. Первая – для атак на распределенные ИС, вторая – на локальные ИС. Основной особенностью распределенной системы является то, что ее компоненты распределены в пространстве, и связь между ними физически осуществляется при помощи сетевых соединений и программно при помощи механизма сообщений. При этом все управляющие сообщения и данные, пересылаемые между объектами распределенной вычислительной системы, передаются по сетевым соединениям в виде пакетов обмена. Эта особенность и является основной для рассматриваемых удаленных атак на инфраструктуру и протоколы распределенных систем. В связи с тем что не было найдено научных исследований, в которых проводилось бы различие между локальными и удаленными информационными воздействиями на ИС, применение уже известных обобщенных классификаций для описания удаленных воздействий не позволяет наиболее точно раскрыть их сущность и условия их осуществления. Это связано с тем, что данный класс воздействий характеризуется сугубо специфичными признаками для распределенных вычислительных систем. Поэтому для описания удаленных атак предлагается следующая классификация:

1. По характеру воздействия:
– пассивные;
– активные.
2. По цели воздействия:
– нарушение конфиденциальности информации либо ресурсов системы;
– нарушение целостности информации;
– нарушение доступности системы.
3. По условиям начала осуществления воздействия:
– атака по запросу от атакуемого объекта;
– атака по наступлению ожидаемого события на атакуемом объекте;
– безусловная атака.
4. По наличию обратной связи с атакуемым объектом:
– с обратной связью;
– без обратной связи.
5. По расположению субъекта атаки относительно атакуемого объекта:
– внутрисигментное;
– межсегментное.

6. По уровню модели ISO / OSI, на котором осуществляется воздействие:

– физический;
– канальный;
– сетевой;
– транспортный;
– сеансовый;
– представительный;
– прикладной.

Исследования подтверждают тот факт, что независимо от используемых протоколов, топологии, инфраструктуры распределенных вычислительных систем механизмы реализации удаленных воздействий инвариантны по отношению к особенностям конкретной системы. Это объясняется тем, что распределенные вычислительные системы проектируются на основе одних и тех же принципов, а следовательно, имеют практически одинаковые проблемы безопасности. Таким образом, появляется возможность использования понятия «типовая удаленная атака». Одна из методик обеспечения безопасности, основанная на типовых удаленных атаках, заключается в последовательном осуществлении всех типовых удаленных воздействий с последующей оценкой защищенности системы. Различными исследователями предлагается следующий набор типовых механизмов воздействий:

1. Анализ сетевого траффика.

2. Подмена доверенного объекта или субъекта распределенной вычислительной системы:

– атака при установленном виртуальном канале;
– атака без установленного виртуального канала.

3. Ложный объект распределенной вычислительной системы:

– использование для организации удаленной атаки на распределенную вычислительную систему;
– селекция потока информации и сохранение ее на ложном объекте распределенной вычислительной системы.

4. Модификация информации:

– модификация передаваемых данных;
– модификация передаваемого кода;
– внедрение разрушающих программных средств;
– изменение логики работы исполняемого файла;
– подмена информации.

5. Отказ в обслуживании.

Локальные атаки совершаются в ИС, которые по своей структуре состоят из нескольких уровней, определяемых как:

а) уровень прикладного ПО, отвечающий за взаимодействие с пользователем. Примером элементов ИС, работающих на этом уровне, можно назвать текстовые редакторы, редакторы электронных таблиц, почтовую программу и т. д.;

б) уровень системы управления базами данных (далее СУБД), отвечающий за хранение и обработку данных информационной системы. Примером элементов ИС, работающих на этом уровне, является СУБД Oracle, MS SQL Server, Sybase и даже MS Access;

в) уровень операционной системы (ОС), отвечающий за обслуживание СУБД и прикладного программного обеспечения. Примерами элементов ИС, работающих на этом уровне, могут служить ОС Microsoft Windows NT, Sun Solaris, Novell Netware;

г) уровень сети, отвечающий за взаимодействие узлов информационной системы. Примером элементов ИС, работающих на этом уровне, можно назвать протоколы TCP / IP, IPS / SPX и SMB / NetBIOS.

Сами атаки классифицируются следующим образом:

– несанкционированный доступ к паролю и конфиденциальной информации;
– нарушение прав доступа;
– атаки типа «отказ в обслуживании»;
– загрузка враждебного содержания (программ типа «троянский конь», мобильного кода Java и ActiveX, вирусов).

Типовыми способами реализации локальных атак являются:

1) Неконтролируемый запуск программ:

а) в непредусмотренных ситуациях (переполнение буфера; неверная обработка системных ситуаций; непредусмотренные входные данные);

б) наличие люков (недокументированные вызовы и флаги; возможность отладки программ или процессов);

в) подмена: данных или программ в оперативной памяти; специальных управляющих переменных; файлов через ссылки или синонимы.

2) Наличие анонимного пользователя.

3) Человеческий фактор (слабые пароли; ошибки администрирования).

4) Совместимость с другими ОС:

– нестойкие криптоалгоритмы;
– обратно совместимые функции шифрования;
– наличие диалектов протоколов.

5) Использование общедоступных ресурсов.

6) Неконтролируемое использование модемов и других аппаратных средств.

7) Доступ к информации во временных файлах программ.

Система, основанная на исправлении ошибок в ПО, обнаружении сигнатур сетевых атак или вирусов, не сможет обнаружить новые ошибки в ПО, сетевые угрозы и новые вирусы, действующие по новой технологии, в чем мы часто убеждаемся, узнавая в новостях об очередной «эпидемии» вирусов. Обычно нам сообщается что были обнаружены новая уязвимость и вредоносная программа, которая ее использует для своего распространения, что вирус начал атаковать компьютеры в Интернете и распространяется с большой скоростью. Таким образом, существующие методы защиты не выполняют своих функций в полной мере и нуждаются в дополнении средствами защиты от еще не известных уязвимостей.

Но уязвимость не может быть неизвестной по своей природе. Если уязвимость компьютерной системы – это некая ее неудачная характеристика, которая делает возможным возникновение угрозы, то при отсутствии информации о неудачной характеристике не возникает и не может возникнуть угроза сама по себе. Поэтому рассмотрение уязвимостей как базовой характеристики для построения какой бы то ни было системы обеспечения информационной безопасности не должно использоваться потому, что априори не сможет обеспечить полной защиты. Следовательно, необходима другая базовая единица анализа информации аудита для построения более надежных средств защиты информации. Как мы уже выяснили, это не может быть уязвимость или риск уязвимости. Это должна быть ситуация, приводящая к появлению уязвимостей. Общая последовательность событий до реализации атаки на ИС заключается в поиске уязвимостей, реализации механизма, использующего найденную уязвимость, и совершение атаки путем использования реализованного механизма. Первые два этапа невозможно ограничить, поскольку исследование ИС на предмет уязвимостей – это повседневная работа специалистов в области защиты информации от злоумышленников. Эти этапы совершаются на моделях реальных систем, не контролируются, не могут и не должны контролироваться третьей стороной, поскольку подобный контроль ограничит законные права граждан на свободу. Единственный этап, на котором мы можем контролировать и ограничивать злоумышленников, – это этап совершения атаки с использованием реализованного механизма. Мы не знаем, какая уязвимость используется, но можем перечислить все возможные физические способы доступа к системе и воздействия, которые могут быть совершены во время этого доступа. По сути, мы должны абстрагироваться от понятия угрозы и считать ее неизвестной, скрытой для анализа и рассматривать исключительно воздействия, риски воздействий и сопутствующие характеристики этих воздействий.

Можно выделить три общие стратегии, связанные с принятием решения в условиях риска воздействия:

1. Избежание риска. Состоит в полном ограничении соответствующего воздействия. Данная стратегия приводит к постепенному ограничению функций ИС и в конечном счете полной потере ее функциональности, так как любая функциональность связана с определенным уровнем риска воздействия.

2. Принятие риска воздействия. Данная стратегия приводит к постоянным колебаниям значений показателей информационной безопасности в соответствии с процессом совершения воздействий с использованием новых механизмов атак и защитой от них по мере появления исправлений к ПО.

3. Управление риском. Предполагает идентификацию, определение, оценку и разработку методов управления риском воздействия.

Примечания

1 *Ефимов А.И.* Проблема технологической безопасности программного обеспечения систем вооружения // Безопасность информационных технологий. 1994. № 3–4. С. 22–33.

2 *Ефимов А.И., Ухлинов Л.М.* Методика расчета вероятности наличия дефектов диверсионного типа на этапе испытаний программного обеспечения вычислительных задач // Вопросы защиты информации. 1995. № 3 (30). С. 86–88.

3 *Казарин О.В.* О создании информационных технологий, исходно ориентированных на разработку безопасного программного обеспечения // Вопросы защиты информации. 1997. № 1–2 (36–37). С. 9–10.

С.В. Кудинов

О СОВРЕМЕННЫХ ПРОБЛЕМАХ В ОБЛАСТИ ЭКОНОМИКИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Изложены современные проблемы в сфере экономики информационной безопасности. Особое внимание уделяется практическому аспекту приложения результатов исследований в области экономики безопасности в организациях. В частности, анализируется метод оценки проектов с использованием взвешенно-факторных моделей. На основе проведенного анализа литературы по данному направлению показаны особенности метода и пути его совершенствования с позиции внедрения в организации.

Ключевые слова: экономика информационной безопасности, инвестиционный анализ, взвешенно-факторные модели, проблемы информационной безопасности, оценка ИБ.

Экономика информационной безопасности в организации

Обеспечение и реализация проектов информационной безопасности (ИБ) требуют всего комплекса ресурсов, доступных в организации: денежных, материальных, интеллектуальных и временных. Более того, часто необходимо дополнительное привлечение внешних ресурсов: проектировщиков-архитекторов ИБ, консультантов-внедренцев, разработчиков программного обеспечения, поставщиков оборудования. Подобная усложненность проектов ИБ в техническом и ресурсном планах приводит к тому, что выгоды и эффективность от решений ИБ не всегда оказываются очевидными, особенно для бизнес-пользователей, не сведущих в специфике ИБ. Таким образом, имеют место две крайне актуальные и слож-

ные задачи, связанные с принятием решений по проектам ИБ. Во-первых, необходимо определиться с процессом принятия решений; во-вторых, нужно понять, сколько и почему тратить на ИБ, т. е. подобрать и корректно внедрить правильные методы экономической оценки ИБ. Последний аспект непосредственно зависит от оценки различных решений и ситуаций, связанных с проектами ИБ, и является областью исследования относительно новой науки в ИБ – экономики ИБ (ЭИБ).

ЭИБ исследует экономические аспекты информационной безопасности, а именно, изучает зависящие от нее выгоды и затраты. Экономика ИБ решает три задачи[1]: при условии заданного уровня ИБ (выраженного через какие-либо показатели) определить минимальный достаточный бюджет для его достижения в текущих условиях; при условии заданного фиксированного бюджета на ИБ найти оптимальный набор ИБ-проектов в контексте деятельности компании; при условии статус-кво (консенсуса) между бюджетом на ИБ и требуемым уровнем ИБ определить необходимые изменения в программных комплексах и оборудовании, которые улучшают уровень ИБ при оптимальных финансовых затратах. Капитальные вложения в ИБ стали основной темой с постоянно возрастающим интересом к ней в последние годы[2]. Один из ее аспектов касается необходимости рассматривать ИБ-инвестиции в свете рациональных экономических терминов аналогично традиционным вложениям. Однако задача определения бизнес-стоимости не стала проще с внедрением массивных корпоративных систем или увеличивающейся значимостью программных и аппаратных средств. В конечном счете мнение о ИБ-проекте существенным образом зависит от предубеждений аналитика и его критической оценки входных данных[3].

К сожалению, в настоящее время представлено достаточно ограниченное число исследований, которые бы имели практический (оперирование концепциями и эмпирические исследования), нежели теоретический, смысл (формулирование предложений)[4]. Следует также отметить, что, несмотря на наличие в литературе достаточно богатого спектра методов экономической оценки ИБ, крайне мало внимания уделяется адаптации их к конкретной ситуации. В отечественной академической литературе практически не освещаются задачи в этой мультидисциплинарной области, стоящей на стыке финансов, управления рисками и информационной безопасности, а тем более – прикладные аспекты внедрения теоретических методов в организации. Таким образом, приобретает актуальность задача не просто выбора метода, но и проверки его применимости на практике.

Проблемы экономического измерения ИБ

Проблемы, связанные с современными методами оценки ИБ-проектов, можно разбить на три группы: надежность данных (исторические данные редко помогают в широкой области ИБ предсказать будущее); оцифровка данных (большинство современных методик оценки рисков используют качественный подход); сложность (область ИБ необычайна сложна)[5]. Однако в проблеме измеримости ИБ помимо этого существуют несколько принципиальных моментов.

Во-первых, основная разница между обычными и ИБ-инвестициями состоит в сложности определения их экономической пользы, так как выгоды от них заключаются в уменьшении потенциальных рисков. Таким образом, ИБ-инвестиции являются уникальной формой экономии затрат, когда не представляется возможным увидеть ущерб от предотвращенных ими ИБ-вторжений, так как последние не произойдут.

Во-вторых, одна из проблем, возникающих при оценке ИБ-проектов, – сложность определения динамики человеческого поведения при внедрении проекта (например, при внедрении программы обучения в области ИБ обычных пользователей). Очень трудно, если не невозможно, оцифровать «мягкие» условия (зависящие от человеческого фактора)[6]. Субъективной оценке, сравнению или градированию уделено в этой задаче не последнее место, что делает крайне актуальной задачу анализа (например, на устойчивость) разработанных моделей, предлагаемых к использованию, с участием экспертов.

В-третьих, в настоящее время наблюдается реальная нехватка качественной статистики по ИБ, тем более что многие источники таких данных (например, вендоры или исполнительные органы по ИБ) не являются в должной мере независимыми и часто либо недооценены, либо переоценены[7]. Также, например, банки и интернет-провайдеры считаются основными «черными дырами», и данные по ним фрагментированы или попросту недоступны. Для того чтобы быть полезными, данные, включая экономические, должны быть рассмотрены в контексте решений. Однако в отношении ИБ часть допущений строится статистически на малых выборках, а часть данных требует индустриальной специфики, которая недоступна или искажена.

В-четвертых, традиционный подход к оценке ИБ провален по ряду причин, основной из которых является гигантская сложность больших компьютерных систем. Анализ затрат/выгод ИБ теряет свою важность в случаях со сложными проблемами в области ИБ,

затрагивающими почти все аспекты бизнеса, например в случае проектов по обеспечению непрерывности бизнеса (BCP – business continuity planning)[8].

Таким образом, любой подход к экономической оценке ИБ должен содержать «противоядия» против названных четырех преград: незаметности ИБ в создании добавленной стоимости, неизбежного вовлечения высокой доли субъективности в анализ, отсутствия должных статистической и исторической баз, а также сложности и специфики увязки между многочисленными компонентами ИБ.

Взвешенно-факторные модели

Как правило, каждый проект характеризуется определенным рядом интегральных показателей, которые заносятся в формальную модель принятия решения. Одной из преследуемых целей является исключение определенного произвола, характерного для качественных методов, с целью получения измеримых и предсказуемых результатов, выраженных в денежных и количественных величинах, понятных для широкого круга людей. Также формальные инструменты призваны учесть и сокрыть специфичность ИБ, выдавая на выходе сопоставимые денежные и количественные показатели. Ключевым элементом количественного подхода является методика оцифровки и анализа проектов, а также ее реализующая модель. В настоящее время разработано множество подходов к оценке ИБ: фиксирование бюджета и использование эталонов; стоимостные методы; способы оценки с применением мер рисков; расчет показателей возврата от инвестиции; классический инвестиционный анализ; методы расчета возврата на атаку и теория игр; рыночные подходы к оценке ценности ИБ; взвешенно-факторные модели (Weighted factor scoring models – WSFM); специализированные (авторские) подходы.

Проведенное в 2003 г. исследование показало, что 92% организаций воспринимают ИБ как чистые затраты в бухгалтерском смысле и 48% из них относятся к ИБ как к инвестициям[9]. Однако ни один из опрошенных не подразумевал расчет показателей «возврата инвестиций в безопасность». Вместо этого толкование ИБ ранжировалось, начиная с того, что ИБ есть необходимая форма уменьшения риска (защиты от риска), и заканчивая тем, что ИБ – источник конкурентного преимущества. Таким образом, вложения в ИБ движутся именно риск- и бизнес-целями. В частности, подход к оценке ИБ часто подразумевает не столько какой-либо расчет, сколько суждение, основанное на здравом смысле и убеждениях в

отношении риска или бизнеса задачи. В отличие от альтернативных методов, WFSM признают наличие высокой субъективности принятия решений в реальной жизни и необходимость учета разносторонних мнений и участников ИБ-процесса. Заметим, что WFSM применяются в областях управления рисками, информационных технологий, проектного управления. Известными авторами в области ЭИБ рекомендуется применять именно многокритериальные системы (типа WFSM) для задач выбора ИБ-решений[10].

В использовании WFSM следует отметить их простоту, гибкость и понятность результатов и расчетов в основе моделей. Так, например, известная модель инвестиционной оценки для ИБ Multi-attribute Analysis (MAA)[11], являющаяся разновидностью WFSM, строится следующим образом: определение факторов, важных для принятия решения; их ранжирование при помощи экспертных оценок; анализ технологий по факторам с привлечением компетентных специалистов; получение общего ранга для каждой технологии с использованием аддитивной модели; ранжирование технологий; анализ расхождения мнения эксперта до и после оценивания. Баланс между простотой и отражением существенных моментов выгодно отличают модели WFSM от их альтернатив: WFSM позволяют эффективно фильтровать факторы и базируются на системе взвешивания, основанной на здравом смысле. Более того, у WFSM есть уникальная возможность «подстраивания» под требования и потребности людей, принимающих окончательное решение (проектных спонсоров). В частности, WFSM можно гибко адаптировать, добавляя или удаляя из них методы в зависимости от конкретной ситуации. Далее, WFSM позволяют агрегировать результаты анализа, полученные другими методами оценки ИБ-инвестиций. Такой подход также дает возможность размывать неточности одиночных оценок, при этом сохраняя рассмотрение инвестиции в ИБ с различных точек зрения. И наконец, WFSM предполагают вовлечение всех заинтересованных сторон. Например, люди, принимающие решения, обеспечивают определение факторов и их весомость, а эксперты из различных областей осуществляют оценку ИБ-решения по факторам.

Высказанное объясняет, почему метод взвешенных оценок в настоящее время наиболее часто применяется в прикладных задачах, связанных с субъективной оценкой различных рисков. Осталось только научиться корректно применять эти модели для оценки инвестиций в ИБ на практике. Именно практический аспект WFSM сопряжен с рядом моментов, которые специалисты ИБ вынуждены решать самостоятельно (так как ни стандарты безопасности, ни ака-

демические работы их не рассматривают): отсутствие оптимального метода ранжирования и выбора факторов/весов, а также методологии внедрения в организации, разграничивающей выгоды для участников процесса. Однако даже если решены означенные вопросы, применение конкретной разновидности WFSM может не привести к ожидаемым результатам[12]. Например, это может быть связано с тем, что при реализации подходов, использующих экспертную оценку и взвешивание для получения единого показателя, оценки на номинальной шкале часто воспринимаются как оценки на шкале отношений, а очки, принадлежащие разным областям, складываются вместе, как если бы они были взаимозаменяемы[13].

Таким образом, WFSM требуют анализа адекватности для применения в конкретной ситуации. К сожалению, в настоящее время этот аспект не раскрыт как в отечественной, так и зарубежной литературе. Эффективный процесс анализа адекватности моделей WFSM требует решения двух задач: формализации модели и автоматизации алгоритмов адекватности. Автором настоящей статьи был предложен ряд алгоритмов подобного анализа с использованием понятий устойчивости и чувствительности моделей[14], часть из которых были автоматизированы и применены на практике для анализа конкретных моделей WFSM оценки проектов ИБ.

Заключение

Показано, что в достаточно молодой области экономики информационной безопасности уже предложено много концепций и актуальных задач. Однако в ней много еще предстоит доработать для перехода от теоретических изысканий к практической области. Решения ИБ требуют расширенной и целостной методологии, которая бы не зависела от искусственно созданной необходимости транслировать все в экономические термины, была бы более прозрачной и восприимчивой к внешней критике, прагматично представляла бы интересы каждого заинтересованного лица. Методы, основанные на WFSM, наиболее близко подошли к решению этой задачи. Однако и в отношении них существует ряд проблем, препятствующих их корректному прикладному использованию в организации. В частности, необходим анализ адекватности отражения разработанной модели для корректной оценки инвестиций ИБ с учетом бизнес-интересов конкретной организации.

Примечания

1 *Klemen M., Biffl S.* Economic Aspects and Needs in IT-Security Risk Management for SMEs // Proc. 6th Intern. Workshop on Economics-Driven Software Engineering. 2004. P. 43–47.

2 *Gordon L., Loeb M., Lucyshyn W.* Economic Aspects of Controlling Capital Investments in Cyberspace Security for Critical Infrastructure Assets // Annual Workshop on Economics and Information Security, College Park, MD, 2003.

3 *Mcculloch B.* Running head: open source operational processes // S145976587. onlinehome.us/research/wp-content/uploads/2007/10/b7783mccullochopen-sourcefinal.doc.

4 *Barua A., Kriebel C., Mukhopadhyay T.* MIS and information economics: augmenting rich descriptions with analytical rigor in information systems design // Proc. X Intern. Conf. on Information Systems. N. Y.: ACM New York, 1989. P. 327–339.

5 *Bohme R., Nowey T.* Economic Security Metrics // LNCS 4909 Dependability Metrics. Berlin; Heidelberg: Springer Verlag, 2008. P. 176–187.

6 *Dhillon G.* Interpreting the management of information systems security // PhD thesis. London: London School of Economics and Political Science, 1995.

7 *Anderson R., Bohme R., Clayton R., Moore T.* Security Economics and the Internal Market. ENISA, 2008.

8 *Caruso D.* The Myth of Cost-Benefit Analysis: The US government's method for evaluating risk isn't as objective as it's made out to be // Strategy and Business. 2008. Vol. 50. P. 30.

9 *Ezingeard J., Bowen-Schrire M.* Information Security: A Strategic Issue: A conjoint report study. Hanley Management College, UK and Dataforeningen, 2003.

10 *Arora A., Hall D., Pinto C., Ramsey D., Telang R.* An ounce of prevention vs. a pound of cure: how can we measure the value of IT security solutions?: Paper LBNL-54549. Lawrence Berkeley National Laboratory, 2004.

11 *Butler S, Shaw M.* Incorporating Nontechnical Attributes in Multi-Attribute Analysis for Security // Proc. EDSER-4: Workshop on Economics-Driven Software Engineering Research. 2002.

12 *Кудинов С.В.* Анализ моделей принятия решений по проектам в области информационной безопасности // Вестник РГГУ. 2009. № 10. Серия «Информатика. Защита информации. Математика». С. 178.

13 *Mercken R.* IT Investment Decisions: Value, Uncertainty and Gut Feeling. Tijdschrift voor economie en management. 2005. Vol. 50.

14 *Кудинов С.В.* Методика анализа взвешенно-факторных моделей принятия решений в области информационной безопасности // Сб. научн. докл. X Всерос. симпозиум по прикладн. и промышл. математике (Сочи–Дагомыс, 1–8 окт. 2009 г.) // Обозрение приклад. и промышл. математики. 2009. Т. 16. Вып. 4. С. 673–676.

А.Н. Приезжая

АНАЛИЗ НОРМАТИВНО-МЕТОДИЧЕСКИХ ДОКУМЕНТОВ В ОБЛАСТИ ЗАЩИТЫ ПЕРСОНАЛЬНЫХ ДАННЫХ

При обеспечении безопасности ПДн операторы зачастую сталкиваются с необходимостью выполнения как требований ФСТЭК России, так и требований ФСБ России, что представляет собой нетривиальную задачу. В данной статье предлагается метод разработки модели угроз и возможностей нарушителя, соответствующей требованиям регуляторов. Разработанный автором метод строится на выявлении ключевых понятий и построении на их основе непротиворечивой терминологии и, соответственно, единой модели.

Ключевые слова: персональные данные, нормативные документы, модель угроз, модель нарушителя, требования.

С введением Федерального закона Российской Федерации от 27 июля 2006 г. № 152-ФЗ «О персональных данных» операторы персональных данных (ПДн), а ими в той или иной степени являются все юридические лица, столкнулись с необходимостью обеспечения безопасности ПДн. В ряде случаев (в том числе и для государственных структур) возникает необходимость аттестации по требованиям безопасности информации информационных систем, обрабатывающих ПДн[1]. Иными словами, организация должна выполнить весь комплекс мероприятий по созданию СЗИ, в том числе разработку документации, отвечающей требованиям ФСТЭК России[2] и ФСБ России (в случае применения средств криптографической защиты) как в области защиты ПДн, так и в области защиты информации конфиденциального. Кроме того, ПДн до недавнего времени не выделялись из общего массива информации, обрабатываемой в организации,

что еще больше затрудняет создание СЗИ, отвечающей требованиям регуляторов.

Несмотря на то что нормативно-методическая документация ФСТЭК России и ФСБ России направлена на решение одной задачи – обеспечения безопасности информации, в документах используются различные подходы к обеспечению безопасности, в том числе в них используется разная терминология. Такая ситуация, естественно, ведет к появлению определенных трудностей при разработке документов, в частности модели угроз безопасности ПДн и возможностей нарушителя. Тем не менее практика показывает, что разработать и согласовать модель угроз можно, хотя для этого необходим не совсем стандартный подход: проведенный анализ НМД-регуляторов показал, что критичными для них являются *разные* составляющие модели угроз и нарушителя, что позволяет объединить подходы, используя в качестве основы ключевые абстракции, общие для рассматриваемых подходов.

После издания Федерального закона появилось достаточно много статей, указывающих на слабости и противоречия в существующей нормативной базе, а также большое количество публикаций по проблемам применения документов регуляторов, в первую очередь ФСТЭК России. Необходимо отметить, что большая часть публикаций либо обозначает проблемы[3], либо представляет собой пересказ руководящих документов с комментариями[4], в них, как правило, не предлагаются решения обозначенных проблем[5]. В данной статье предлагается метод разработки модели угроз, соответствующей требованиям регуляторов и адекватной объекту.

Рассмотрим структуру нормативной методической документации (НМД) в области обеспечения безопасности конфиденциальной информации. В первом приближении структура НМД может быть описана следующей схемой (рис. 1).

В соответствии с приведенной схемой нормативная база для разработки системы защиты выбирается в зависимости от вида (видов) обрабатываемой информации и условий ее обработки. При этом в системе одновременно могут обрабатываться несколько видов информации, а использование нормативной базы ФСБ России в общем случае не отменяет обязательность исполнения требований ФСТЭК России.

Каждый регулятор создает собственную нормативно-методическую базу, используя различную терминологию и методический подход, более того, терминология и подход, используемые регуляторами при разработке методических рекомендаций по защите ПДн, несколько отличаются от таковых, используемых в НМД по

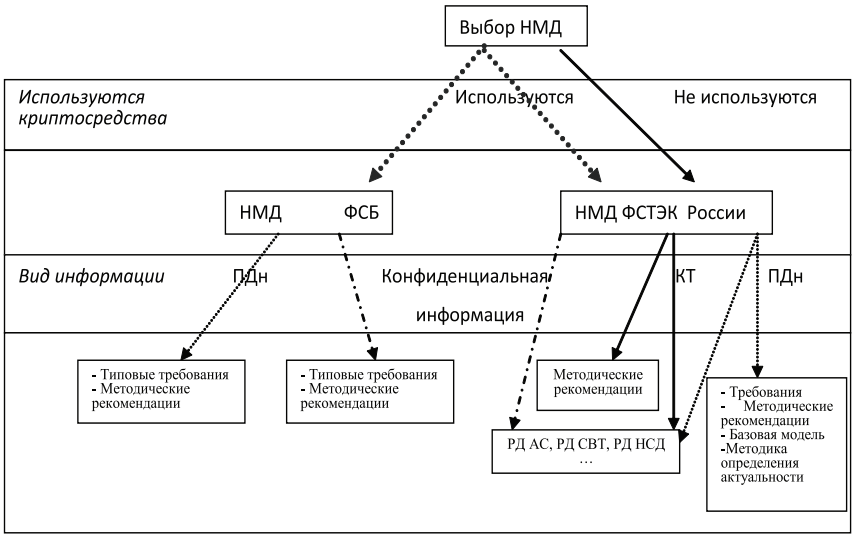


Рис. 1. Схема НМД в области обеспечения безопасности конфиденциальной информации

защите информации конфиденциального характера. В данной статье проводится анализ НМД по защите персональных данных ФСТЭК России и НМД ФСБ России.

Нормативно-методическая документация по обеспечению безопасности ПДн разработана регуляторами в соответствии с Постановлением Правительства Российской Федерации от 17 ноября 2007 г. № 781 «Об утверждении Положения об обеспечении безопасности персональных данных при их обработке в информационных системах персональных данных», и при разработке этих документов изначально предполагалось их совместное использование, но, к сожалению, расхождения в подходах полностью устранены не были.

Требования к обеспечению безопасности ПДн с использованием криптографических средств при их обработке в информационной системе ПДн предъявляются ФСБ России[6], при этом руководящие документы ФСБ России не исключают требований ФСТЭК России, а дополняют их[7]. В руководящих документах используется традиционный для ФСБ подход определения тре-

бований к системе в зависимости от возможностей потенциального нарушителя.

«Методические рекомендации...» выделяют два вида модели угроз: верхнего уровня и детализированную. Модель угроз верхнего уровня предназначена для определения характеристик безопасности защищаемых объектов и используется при построении детализированной модели угроз. Детализированная модель предназначена для определения требуемого уровня криптографической защиты, то есть для определения требований к функциональным возможностям системы защиты информации.

В контексте модели верхнего уровня ФСБ России определяет угрозу безопасности объекта как возможное нарушение характеристики безопасности объекта, соответственно модель угроз верхнего уровня определяется перечнем всех характеристик безопасности для всех возможных объектов угроз, например угроза модификации информации о состоянии банковского счета.

Для создания детализированной модели угроз необходимо определить совокупность условий и факторов, создающих опасность нарушения характеристик безопасности возможных объектов угроз. Понятие угрозы безопасности в контексте детализированной модели угроз отличается от угрозы безопасности верхнего уровня. В частности, разделяются атаки[8] и угрозы, не являющиеся атаками, что подразумевает включение в понятие угрозы методов ее реализации. При этом модель угроз разрабатывается через перечень возможностей нарушителя[9], то есть подразумевается, что угрозы, не являющиеся атаками, из рассмотрения фактически убираются[10]. Иными словами, детализированная модель угроз должна содержать максимально полное описание атак. Атака как любое целенаправленное действие характеризуется рядом существенных признаков: субъект атаки (нарушитель), объект атаки, цель атаки, имеющаяся у нарушителя информация об объекте атаки, имеющиеся у нарушителя средства атаки, канал атаки.

Возможные объекты атак и цели атак определяются на этапе формирования модели угроз верхнего уровня. Как было сказано выше, возможность проведения атак обусловлена возможностями нарушителя, то есть перечень возможных атак определяется моделью нарушителя. В силу этого утверждения ФСБ России предъявляет требования к информационным системам, обрабатывающим ПДн, в зависимости от типа нарушителя (Н1, Н2, ... , Н6) (табл. 1).

Таблица 1

Тип нарушителя / Возможность	Н1	Н2	Н3	Н4	Н5	Н6
Возможность сговора						
Известны все сети связи, работающие на едином ключе						
Располагают исходными текстами прикладного программного обеспечения						Располагают всей документацией на криптосредство и СФК
Располагают не только доступными в свободной продаже аппаратными компонентами криптосредства и среды функционирования криптосредства (СФК)			В зависимости от реализованных в ИС организационных мер			Любыми компонентами
Использование штатных средств	Только если они расположены за пределами КЗ	В зависимости от реализованных в ИС организационных мер				
Могут проводить лабораторные исследования криптосредств, используемых за пределами контролируемой зоны ИС						
Могут ставить работы по созданию способов и средств атак в научно-исследовательских центрах, специализирующихся в области разработки и анализа криптосредства и СФК						

ФСБ России определяет следующую структуру модели нарушителя безопасности ПДн:

– описание нарушителей (возможности внешних и внутренних нарушителей, возможности сговора, тип нарушителя, категории лиц – потенциальных нарушителей);
– предположения об имеющейся у нарушителя информации об объектах атак;

– предположения об имеющихся у нарушителя средствах атак;
– описание каналов атак (каналы связи, штатные средства, носители информации, каналы непосредственного доступа к объекту атаки, технические каналы утечки).

Термины, использованные в руководящих документах ФСБ России (рис. 2), в значительной степени пересекаются с терминами документов ФСТЭК России[11–14], однако далеко не всегда эти термины используются в точности в том же значении, что и порождает трудности для операторов (и/или интеграторов), пытающихся реализовать одновременно требования ФСТЭК и ФСБ.

Таким образом, ФСБ России предлагает такой порядок разработки модели угроз (рис. 3).

Классификация ИСПДн осуществляется на основе модели нарушителя, требования к функциональным возможностям средств защиты предъявляются только в части, касающейся применения криптографических средств.

Документы ФСТЭК России, регламентирующие защиту ПДн, к сожалению, нуждаются в доработке. Тем не менее строить системы защиты, отвечающие требованиям данных документов, приходится уже сейчас.

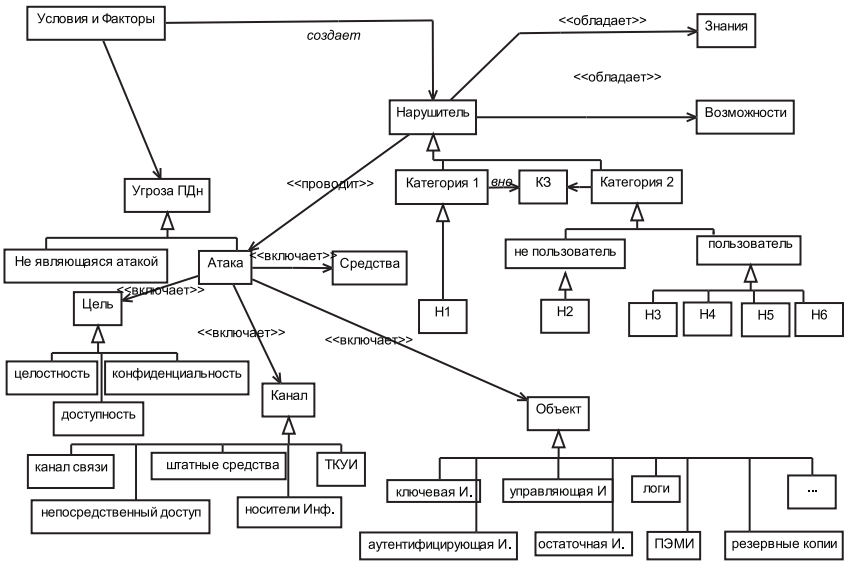


Рис. 2. НМД ФСБ

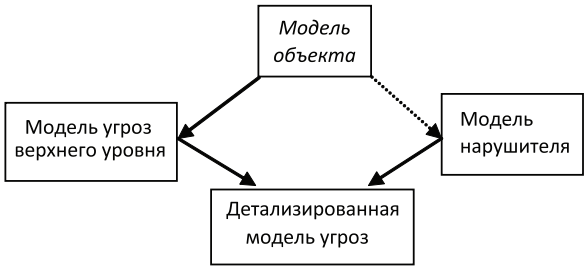


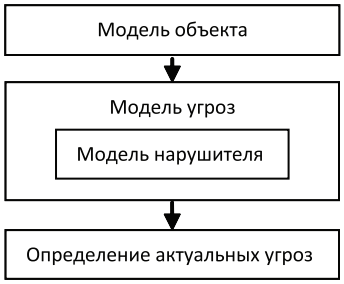
Рис. 3. Порядок разработки модели угроз безопасности ПДн

Основой для классификации информационных систем и построения системы защиты ПДн в соответствии с РД ФСТЭК России является модель актуальных угроз. Возможность реализации угрозы определяется условиями и факторами, а ее реализация влечет за собой последствия, которые необходимо учитывать при определении актуальности угроз. Актуальность угроз зависит от исходной степени защищенности автоматизированной системы; вероятности реализации угрозы; показателя опасности угрозы. Вероятность реализации угрозы и степень ее опасности определяются экспертным путем. Данный подход включает в себя элементы анализа рисков, в то время как с точки зрения ФСБ России модели угроз полностью определяются возможностями нарушителя.

В терминологии ФСТЭК России угроза включает в себя источник угрозы, уязвимость, способ реализации угрозы, объект воздействия, несанкционированный доступ и реализуется через канал реализации (источник угрозы, среда распространения, носитель информации). Источником угрозы могут быть нарушитель[15], закладное устройство[16], вредоносная программа. Возможности источников определяются методами и средствами несанкционированного доступа. При этом в документах ФСБ России закладные устройства и вредоносное ПО не рассматриваются как самостоятельный источник угрозы, так как их внедрение обусловлено действиями нарушителя или ошибочными действиями пользователя. Здесь необходимо отметить, что ФСБ России определяет нарушителя как лицо (или процесс), проводящее атаку, и рекомендует исключать из числа нарушителей доверенных лиц (например, пользователей группы администраторов), а в терминах ФСТЭК России нарушителем является лицо, совершающее не только преднамеренные

(атаки), но и случайные действия, приводящие к нарушению безопасности информации, в частности, «Базовая модель...» ФСТЭК России предлагает такие категории нарушителей, как зарегистрированные пользователи с полномочиями администратора безопасности автоматизированной системы и зарегистрированные пользователи с полномочиями системного администратора автоматизированной системы.

Разработка модели угроз в соответствии с руководящими документами ФСТЭК России осуществляется по следующей схеме:



В данной схеме модель нарушителя не является самостоятельным документом, она представляет собой второстепенную часть модели угроз, а модель угроз имеет один уровень детализации, соответствующий детализированной модели угроз в терминологии ФСБ России. Основные термины и понятия, используемые при разработке модели угроз ПДн в соответствии с РД ФСТЭК, приведены на рис. 4.

Из всего вышесказанного следует, что руководящие документы ФСТЭК И ФСБ России предлагают несколько различный подход к разработке модели угроз и используют разную терминологию, в частности не совпадают такие основополагающие понятия, как «нарушитель» и «угроза». Согласно предлагаемому регуляторами подходу, классификация автоматизированных систем и определение функциональных требований к СЗИ должны осуществляться на основании модели угроз, которую рекомендуется согласовывать с регуляторами.

При объединении двух подходов необходимо найти совпадающие абстракции, в данном случае рассматриваются понятия «объект воздействия» и «объект атаки», обозначающие идентичные сущности (данные понятия связаны соответственно с угрозой и атакой), далее определяется примерное соответствие терминов ФСТЭК и ФСБ России (табл. 2):

Таблица 2

ФСБ России	ФСТЭК России	
Атака	Угроза	
Нарушитель	Источник	
Средства	Методы и средства	
Цель	НСД	
Канал атаки	Способ реализации	Канал реализации
		Уязвимость
Нет соответствий	Вероятность реализации	
	Показатель опасности	

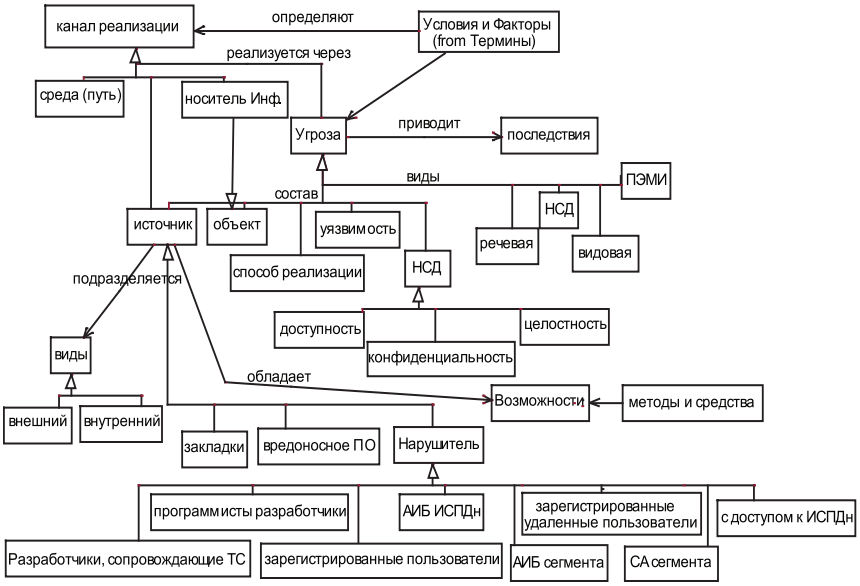


Рис. 4. НМД ФСТЭК

Фактически руководящие документы ФСТЭК России требуют разработки перечня актуальных угроз вида: источник угрозы, способ реализации, объект воздействия, несанкционированный доступ, уязвимость. Модель нарушителя не играет никакой роли в классификации систем и используется только в процессе формирования перечня угроз[17], что позволяет классифицировать и описывать нарушителя в соответствии с требованиями ФСБ России. Категории нарушителя, приведенные в РД ФСТЭК России, будут использованы в процессе определения категорий лиц, которые могут являться потенциальными нарушителями, и их возможностей. При этом в результирующей модели угроз необходимо описать не только атаки, то есть угрозы, умышленно реализованные нарушителем, но и угрозы, связанные с непредумышленными (ошибочными) действиями санкционированных пользователей, приводящими к нарушению безопасности информации. Угрозы, источником которых являются вредоносные программы и аппаратные закладки, целесообразно рассматривать в рамках ошибочных действий пользователя или атак, проводимых нарушителем, т. е. программно-математическое воздействие и внедрение аппаратных закладок можно рассматривать либо как метод реализации атаки, либо как самостоятельную угрозу, источником которой является нарушитель, что позволит, не исключая данные угрозы из рассмотрения, придерживаться при построении модели нарушителя терминологии ФСБ России. Выбор терминологии и методологии ФСБ России для данного раздела обусловлен необходимостью классификации нарушителя на основании руководящих документов ФСБ России. После определения возможностей нарушителя и способов реализации атак (перечня угроз), необходимо провести экспертную оценку вероятности реализации угрозы (атаки) и степень ее опасности для конкретного ресурса системы.

Таким образом, разрабатывается модель нарушителя, позволяющая создать классификацию системы и выбрать необходимый уровень криптографической защиты в соответствии с РД ФСБ России, и перечень актуальных угроз (способов реализации), позволяющий определить требуемый уровень защиты автоматизированной системы от НСД и утечки по техническим каналам, в соответствии с РД ФСТЭК России.

В качестве первого раздела модели целесообразно включать описание ИС как объекта защиты (в частности, назначение, архитектура и структура ИС, состав, обрабатываемая информация и пользователи ИС, функциональные возможности ИС, информационное взаимодействие подсистем ИС).

Таким образом, используя предложенный метод разработки модели угроз безопасности информации и потенциальных

возможностей нарушителя, можно выполнить требования регуляторов в области технической защиты информации, что в свою очередь обеспечивает согласованность документов.

Примечания

1 *Сабанов А.Г.* Некоторые проблемы защиты персональных данных // Бизнес и безопасность в России. 2009. № 52. С. 37–38.

2 *Назаров И.Г.* Основные вопросы практической реализации требований по обеспечению безопасности персональных данных при их обработке в информационных системах // Там же. С. 13–16.

3 См.: *Голованова Е.* Защита персональных данных: проблемы операторов // Information Security / Информационная безопасность. 2008. № 5; *Волчинская Е.К.* Некоторые правовые проблемы применения Федерального закона «О персональных данных» // Персональные данные. 2009. № 2.

4 См.: *Коржов В.* Защита персональных данных: проблемы и решения [Электронный ресурс] // Сайт «Открытые системы». [М., 2009]. URL: http://www.osp.ru/os/2009/06/10050193/ (дата обращения: 4.11.2009).

5 См.: Выполнимы ли требования Федерального закона «О персональных данных»: единого мнения нет [Электронный ресурс] // Сайт «Портал персональных данных». [М., 2009]. URL: http://pd.rsoc.ru/press-service/subject3/news304.htm (дата обращения: 4.11.2009).

6 Методические рекомендации по обеспечению с помощью криптосредств безопасности персональных данных при их обработке в информационных системах персональных данных с использованием средств автоматизации. Утверждены руководством 8 Центра ФСБ России 21 февраля 2008 г. № 149/5-144. Типовые требования по организации и обеспечению функционирования шифровальных (криптографических) средств, предназначенных для защиты информации, не содержащей сведений, составляющих государственную тайну, в случае их использования для обеспечения безопасности персональных данных при их обработке в информационных системах персональных данных. Утверждены руководством 8 Центра ФСБ России 21 февраля 2008 г. № 149/6/6-622.

7 При этом из двух содержащихся в документах ФСТЭК России и ФСБ России однотипных угроз выбирается более опасная.

8 Атаки готовятся и проводятся нарушителем, причем возможности проведения атак обусловлены возможностями нарушителя.

9 Нарушитель (субъект атаки) – лицо (или инициируемый им процесс), проводящее (проводящий) атаку.

10 Защита от угроз, не связанных с действиями нарушителя, должна регламентироваться инструкциями.

[11] Рекомендации по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных. Утверждены ФСТЭК России 15 февраля 2008 г.

[12] Основные мероприятия по организации и техническому обеспечению безопасности персональных данных, обрабатываемых в информационных системах персональных данных. Утверждены ФСТЭК России 15 февраля 2008 г.

[13] Методика определенных угроз безопасности персональных данных при их обработке в информационных системах персональных данных. Утверждена ФСТЭК России 14 февраля 2008 г.

[14] Базовая модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных. Утверждена ФСТЭК России 15 февраля 2008 г.

[15] Нарушитель – физическое лицо, случайно или преднамеренно совершающее действия, следствием которых является нарушение безопасности ПДн при их обработке техническими средствами и информационной системы ПДн.

[16] При этом при классификации угроз по видам источников выделяются следующие классы угроз: 1) угрозы, связанные с преднамеренными или непреднамеренными действиями лиц, имеющих доступ к ИСПДн, включая пользователей (внутренний нарушитель); 2) угрозы, связанные с преднамеренными или непреднамеренными действиями лиц, имеющих доступ к ИСПДн (внешний нарушитель).

[17] Классификация нарушителя и угроз носит справочный характер и предназначена для проведения всестороннего анализа системы.

Э.Р. Бейбутов

МЕТОДИКА ОЦЕНКИ СООТВЕТСТВИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ОПЕРАТОРОВ ПЕРСОНАЛЬНЫХ ДАННЫХ ТРЕБОВАНИЯМ ЗАКОНОДАТЕЛЬСТВА В ОБЛАСТИ ЗАЩИТЫ ПЕРСОНАЛЬНЫХ ДАННЫХ

Федеральный закон от 27 июля 2006 г. № 152-ФЗ «О персональных данных» возложил на организации, осуществляющие обработку персональных данных, ответственность за соблюдение требований безопасности любой информации, относящейся к определенному или определяемому на основании такой информации физическому лицу (субъекту персональных данных). Принятые на основании указанного Федерального закона нормативно-правовые акты дополнили и уточнили требования информационной безопасности, создав теоретико-практическую базу проведения работ по защите персональных данных. В настоящей статье предлагается методика, позволяющая решить актуальную проблему оценки полноты выполнения требований законодательства в области защиты персональных данных. Автором проведен анализ нормативно-правовых актов, предложена систематизация требований безопасности по трем уровням: менеджмент, организационный, программно-технический, с дальнейшей их декомпозицией, позволяющей провести количественную и качественную экспертную оценку соответствия операторов персональных данных требованиям законодательства в области защиты персональных данных.

Ключевые слова: законодательство в области персональных данных, защита персональных данных, требования безопасности информации, критерии выполнения требований.

С момента вступления в силу Федерального закона от 19 декабря 2005 г. № 160-ФЗ «О ратификации Конвенции Совета Европы о защите физических лиц при автоматизированной обработке персональных данных» органами законодательной и исполнительной власти Российской Федерации были разработаны и приняты нормативно-правовые акты, раскрывающие обширный пере-

чень обязанностей организаций, осуществляющих обработку персональных данных (ПДн), а также задающих цели и содержание обработки ПДн (далее – операторов ПДн, операторов). Принятое законодательство защищает права и свободы человека и гражданина России, возлагая ответственность на операторов за обеспечение информационной безопасности (ИБ) ПДн при их обработке как автоматизированными, так и неавтоматизированными способами. К основным нормативно-правовым актам, раскрывающим обязанности операторов и порядок проведения мероприятий по защите ПДн, можно отнести следующие:

- Федеральный закон от 27 июля 2006 г. № 152-ФЗ «О персональных данных»;
- Постановление Правительства РФ от 17 ноября 2007 г. № 781 «Об утверждении Положения об обеспечении безопасности персональных данных при их обработке в информационных системах персональных данных»;
- Постановление Правительства РФ от 15 сентября 2008 г. № 687 «Об утверждении Положения об особенностях обработки персональных данных, осуществляемой без использования средств автоматизации»;
- Приказ Федеральной службы по техническому и экспортному контролю (ФСТЭК) России, Федеральной службы безопасности (ФСБ) России и Министерства информационных технологий и связи России от 13 февраля 2008 г. № 55/86/20 «Об утверждении Порядка проведения классификации информационных систем персональных данных»;
- методические рекомендации и типовые требования по обеспечению с помощью криптосредств безопасности персональных данных, принятые 8 Центром ФСБ России 21 февраля 2008 г.;
- основные мероприятия и рекомендации по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных, принятые ФСТЭК России 15 февраля 2008 г.

После проведения анализа требований перечисленных нормативно-правовых актов и практического применения методических документов в создании систем защиты персональных данных (СЗПДн) была выявлена необходимость в методике, позволяющей оценить соответствие ИБ операторов ПДн требованиям законодательства в области защиты ПДн.

Была задана цель работы, которая заключается в создании набора критериев и формировании правил оценки уровня соответствия ИБ операторов ПДн требованиям законодательства в области защиты ПДн.

Методика должна достигнуть поставленной цели при решении следующих задач:

- определение состава критериев и частных показателей ИБ ПДн и правил их оценки;
- определение правил оценивания менеджмента ИБ ПДн;
- определение правил оценивания организационного уровня ИБ ПДн;
- определение правил оценивания программно-технического уровня ИБ ПДн;
- определение итогового уровня соответствия ИБ операторов ПДн требованиям законодательства в области защиты ПДн.

Анализ нормативно-правовых актов позволил распределить требования безопасности информации по трем группам в зависимости от области обеспечения ИБ[1]: требования к уровню менеджмента ИБ ПДн; требования к организационному уровню ИБ ПДн; требования к программно-техническому уровню ИБ ПДн.

Под менеджментом ИБ ПДн будем понимать стратегическое управление обеспечением безопасности ПДн, реализованное путем определения руководством организации целей защиты и принятия решений по их достижению. Менеджмент ИБ ПДн включает в себя следующие процессы:

- определение требований по обеспечению ИБ ПДн на основе анализа моделей угроз и нарушителя ИБ ПДн;
- утверждение способов защиты (мер и средств), позволяющих выполнить требования безопасности информации;
- управление обеспечением (кадровым, финансовым, информационным) для достижения целей защиты ПДн;
- сертификация средств защиты информации и аттестация автоматизированных ИСПДн, а также мониторинг и контроль эффективности внедренных способов защиты;
- соблюдение законности в лицензируемых областях деятельности;
- принятие решений и утверждение планов по улучшению обеспечения ИБ ПДн по результатам анализа эффективности способов защиты или в случаях возникновения инцидентов ИБ ПДн.

Под организационным уровнем ИБ ПДн будем понимать регламентацию технологических процессов обработки ПДн, обслуживания ИСПДн и взаимодействия сотрудников организации на нормативно-правовой основе, исключающей или снижающей вероятность нанесения ущерба правам и свободам субъектов ПДн. На организационном уровне ИБ решаются следующие задачи защиты ПДн:

1) подготовка сотрудников, распределение ролей и назначение ответственности за нарушение безопасности ПДн при их обработке в ИСПДн;

2) применение организационных средств защиты при проектировании, строительстве, оборудовании и эксплуатации помещений, узлов сети и других объектов ИСПДн;

3) разработка частных политик ИБ, регламентирующих управление учетными записями пользователей ИСПДн, внесение изменений в конфигурации программно-технического обеспечения ИСПДн, управление жизненным циклом ПДн и эскалацией инцидентов ИБ ПДн.

Под программно-техническим уровнем ИБ ПДн понимается выполнение задач, поставленных менеджментом организации и на организационном уровне ИБ ПДн требующих проектирования, внедрения и эксплуатации программно-технических средств защиты информации. Полнота выполнения требований программно-технического уровня ИБ ПДн зависит от достижения целей защиты ПДн от несанкционированного доступа (НСД), в том числе с использованием средств криптографической защиты информации, и от утечки по техническим каналам.

Правила выставления баллов по частным показателям

Оценка выполнения требований законодательства в области защиты ПДн рассчитывается по каждой из представленных групп требований. Для этого формируются наборы критериев, а для каждого критерия составляется набор частных показателей. Частные показатели представляются в виде вопросов, ответы на которые ранжируются по следующему принципу[2]:

- «нет» – значению частного показателя присваивается балл, равный нулю;
- «частично» – значению частного показателя присваивается балл, равный 0,25; 0,5 или 0,75;
- «да» – значению частного показателя присваивается балл, равный единице;
- «не применимо» – частный показатель не используется при вычислении оценки выполнения критерия.

Частные показатели каждого критерия имеют весовые коэффициенты значимости, сумма которых равна единице. Если частный показатель предназначен для оценки требований, которые не относятся к деятельности организации или на момент оценки не являются актуальными для организации, что документально зафикси-

ровано во внутренних документах организации, то данный частный показатель определяется как неоцениваемый (должна быть заполнена графа «н/п» – не применимо) и не учитывается в формировании дальнейших результатов оценки. При этом необходимо выполнить процедуру нормировки коэффициентов значимости оставшихся частных показателей ИБ ПДн в рамках критерия.

При выставлении баллов по частным показателям проводится оценка степени документированности требований, или степени выполнения требований, или одновременно степеней документированности и выполнения требований.

При выставлении баллов по частным показателям, для которых оценивается как степень документированности, так и степень выполнения, рекомендуется использовать шкалу, представленную в таблице 1.2. Результаты таблицы 1.2 получены на основе данных таблицы 1.1 и формулы (1).

Таблица 1.1

Рекомендуемые критерии выставления баллов частных показателей ИБ, в которых оценивается как степень документированности, так и степень выполнения требований ИБ

	Не документированы	Частично документированы	Полностью документированы
Не выполняются	0	0	1
Частично выполняются	1	1	2
Полностью выполняются	2	3	4

Для перехода к нормированным оценкам необходимо значение балла разделить на максимально допустимый балл для рассматриваемого частного показателя. В результате вычисляется нормированный балл частного показателя K_{ij}:

$$K_{ij} = \frac{B_{ij}}{B_{ij}^{\max}}, \qquad (1)$$

где B_{ij} – балл, выставленный в соответствии с полнотой реализации требования; $B_{ij}^{\max}$ – максимально допустимый балл для рассматриваемого частного показателя.

Таким образом, таблица 1.1 приводится к виду таблицы 1.2, в которой отражена зависимость нормированных баллов от полноты реализации требования ИБ ПДн.

Таблица 1.2

Рекомендуемые критерии выставления нормированных баллов частных показателей ИБ, в которых оценивается как степень документированности, так и степень выполнения требований ИБ

Нормированный балл частного показателя ИБ	Критерий выставления балла частного показателя ИБ
0	Требования частного показателя ИБ не установлены во внутренних нормативных документах проверяемой организации и не выполняются
0	Требования частного показателя ИБ частично установлены в нормативных документах проверяемой организации, но не выполняются
0,25	Требования частного показателя ИБ полностью установлены в нормативных документах проверяемой организации, но не выполняются
0,25	Требования частного показателя ИБ не установлены во внутренних нормативных документах проверяемой организации и выполняются в неполном объеме
0,25	Требования частного показателя ИБ частично установлены во внутренних нормативных документах проверяемой организации и выполняются в неполном объеме
0,5	Требования частного показателя ИБ полностью установлены во внутренних нормативных документах проверяемой организации и выполняются в неполном объеме
0,5	Требования частного показателя ИБ не установлены во внутренних нормативных документах проверяемой организации, но выполняются в полном объеме
0,75	Требования частного показателя ИБ частично установлены во внутренних нормативных документах проверяемой организации, но выполняются в полном объеме
1	Требования частного показателя ИБ полностью установлены во внутренних нормативных документах проверяемой организации и выполняются в полном объеме

При проведении оценки частных показателей, для которых оценивается только степень документированности, рекомендуется использовать следующий общий подход (табл. 2.1):

Таблица 2.1

Рекомендуемые критерии выставления баллов частных показателей ИБ, в которых оценивается только степень документированности требований ИБ

Не документированы	Частично документированы	Полностью документированы
0	1	2

В соответствии с формулой (1) таблица 2.1 приводится к виду таблицы 2.2, в которой отражена зависимость нормированных оценок от полноты документированности требований ИБ.

Таблица 2.2

Рекомендуемые критерии выставления оценок частных показателей ИБ, в которых оценивается только степень документированности требований ИБ

Нормированный балл частного показателя ИБ	Критерий выставления балла частного показателя ИБ
0	Требования частного показателя ИБ не установлены во внутренних нормативных документах проверяемой организации
0,5	Требования частного показателя ИБ частично установлены в нормативных документах проверяемой организации
1	Требования частного показателя ИБ полностью установлены в нормативных документах проверяемой организации

При проведении оценки частных показателей, для которых оценивается только степень выполнения, рекомендуется использовать следующий общий подход (табл. 3.1):

Таблица 3.1

Рекомендуемые критерии выставления баллов частных показателей ИБ, в которых оценивается только степень выполнения требований ИБ

Не выполняются	Частично выполняются	Полностью выполняются
0	1	2

В соответствии с формулой (1) таблица 3.1 приводится к виду таблицы 3.2, в которой отражена зависимость нормированных оценок от полноты документированности требований ИБ.

Таблица 3.2

Рекомендуемые критерии выставления оценок частных показателей ИБ, в которых оценивается только степень выполнения требований ИБ

Нормированный балл частного показателя ИБ	Критерий выставления балла частного показателя ИБ
0	Требования частного показателя ИБ не выполняются
0,5	Требования частного показателя ИБ выполняются в неполном объеме
1	Требования частного показателя ИБ выполняются в полном объеме

Выставление баллов по частным показателям ИБ должно основываться на свидетельствах аудита, в качестве основных источников которых рекомендуется использовать:

- локальные нормативные документы проверяемой организации и при необходимости документы третьих лиц, относящиеся к обеспечению ИБ ПДн в организации;
- устные высказывания сотрудников проверяемой организации в процессе проводимых опросов;
- результаты наблюдений членов аудиторской группы за деятельностью сотрудников проверяемой организации в области ИБ ПДн.

В процессе проведения устного опроса сотрудников проверяемой организации и наблюдений за деятельностью указанных сотрудников члены аудиторской группы должны сделать вывод о сте-

пени соответствия оцениваемой деятельности требованиям внутренних нормативных документов проверяемой организации.

Полученные свидетельства аудита ИБ ПДн и источники их получения должны быть задокументированы путем составления листов для сбора свидетельств аудита ИБ ПДн. При заполнении листов для сбора свидетельств аудита ИБ ПДн необходимо указать ссылки на соответствующие локальные нормативные документы проверяемой организации, результаты опроса сотрудников проверяемой организации, а также результаты наблюдений членов аудиторской группы. Результаты опроса и наблюдений должны быть подтверждены подписью опрашиваемого сотрудника организации и члена аудиторской группы.

Оценка менеджмента ИБ ПДн

Оценка менеджмента ИБ ПДн осуществляется с помощью критериев и частных показателей ИБ ПДн, позволяющих определить степень выполнения требований законодательства в области ПДн (табл. 4).

Таблица 4

Набор критериев оценки менеджмента ИБ ПДн

Обозначение критерия	Наименование критерия ИБ ПДн
MNG_1	Разработка замысла обеспечения ИБ ПДн
MNG_2	Обоснование требований по обеспечению ИБ ПДн
MNG_3	Разработка внутренних документов, регламентирующих управление обеспечением ИБ ПДн
MNG_4	Разработка программы по обучению и повышению осведомленности персонала в области ИБ ПДн
MNG_5	Выбор целесообразных способов (мер и средств) защиты ПДн в соответствии с задачами и замыслом обеспечения ИБ ПДн
MNG_6	Планирование мероприятий по обеспечению ИБ ПДн
MNG_7	Организация работ по вводу в эксплуатацию СЗПДн в ИСПДн
MNG_8	Мониторинг и контроль эффективности мер обеспечения ИБ ПДн
MNG_9	Аттестация или декларирование соответствия ИСПДн по требованиям безопасности информации
MNG_{10}	Лицензирование деятельности по технической защите конфиденциальной информации
MNG_{11}	Разработка решений по улучшению обеспечения ИБ ПДн

Для каждого критерия менеджмента ИБ ПДн MNG_1–MNG_{11} разрабатываются частные показатели, которым присваиваются весовые коэффициенты значимости. Состав и содержание частных показателей определяются в соответствии с требованиями нормативно-правовых актов в области защиты ПДн. Так, например, для критерия MNG_1 «Разработка замысла обеспечения ИБ ПДн» частными показателями являются[3]:

- определение состава, содержания и местонахождения ПДн, подлежащих защите;
- проведение категорирования ПДн;
- проведение оценки выполнения обязанностей по обеспечению безопасности ПДн оператором;
- проведение оценки возможности физического доступа к ИСПДн;
- выявление возможных технических каналов утечки информации;
- анализ возможностей программно-математического воздействия на ИСПДн;
- анализ возможностей электромагнитного воздействия на ПДн, обрабатываемые в ИСПДн;
- проведение оценки непосредственного ущерба от реализации угроз безопасности ПДн;
- проведение оценки опосредованного ущерба от реализации угроз безопасности ПДн;
- анализ имеющихся в распоряжении мер и средств защиты ПДн от физического доступа;
- анализ имеющихся в распоряжении мер и средств защиты ПДн от утечки по техническим каналам утечки информации;
- анализ имеющихся в распоряжении мер и средств защиты ПДн от несанкционированного доступа;
- анализ имеющихся в распоряжении мер и средств защиты ПДн от программно-математических воздействий;
- анализ имеющихся в распоряжении мер и средств защиты ПДн от электромагнитного излучения;
- определение направлений сосредоточения усилий по защите ПДн;
- выбор основных способов защиты ПДн;
- решение основных вопросов управления защитой ПДн;
- решение основных вопросов финансового, технического и программного, информационного и кадрового обеспечения.

Вычисление оценки по критерию MNG_1 «Разработка замысла обеспечения ИБ ПДн» выполняется по формуле (2.1):

$$MNG_i = \sum_{j=1}^{N} \alpha_{i,j}^{MNG} \times PF_{i,j}^{MNG}, \quad (2.1)$$

где $\alpha_{i,j}^{MNG}$ – весовой коэффициент j-ого частного показателя i-ого критерия менеджмента ИБ ПДн; $PF_{i,j}^{MNG}$ – значение j-ого частного показателя i-ого критерия менеджмента ИБ ПДн, вычисленное по правилам выставления баллов, описанным выше; N – общее количество критериев менеджмента ИБ ПДн.

Аналогичным образом проводится оценка выполнения всех остальных критериев, а для оценки менеджмента ИБ ПДн в целом необходимо определить среднее значение оценок критериев по формуле (3.1):

$$EV_{MNG} = \frac{\sum MNG_i}{N}, \quad (3.1)$$

где EV_{MNG} – оценка менеджмента ИБ ПДн; MNG_i – значение оценки i-ого критерия менеджмента ИБ ПДн.

Оценка организационного уровня ИБ ПДн

Оценка организационного уровня ИБ ПДн осуществляется с помощью критериев и частных показателей ИБ ПДн, позволяющих определить степень выполнения требований законодательства в области защиты ПДн (табл. 5).

Таблица 5

Набор критериев оценки организационного уровня ИБ ПДн[4]

Обозначение критерия	Наименование критерия ИБ ПДн
ORG_1	Подготовка персонала к безопасной обработке ПДн
ORG_2	Обеспечение организационной защиты помещения и средств вычислительной техники
ORG_3	Управление учетными записями
ORG_4	Управление конфигурациями и изменениями
ORG_5	Управление уязвимостями ИСПДн
ORG_6	Управление жизненным циклом обработки ПДн
ORG_7	Управление инцидентами ИБ ПДн
ORG_8	Управление жизненным циклом ключей шифрования СКЗИ
ORG_9	Управление внешними подключениями к ИСПДн

Для каждого критерия организационного уровня ИБ ПДн ORG_1–ORG_9 разрабатываются наборы частных показателей. Каждому частному показателю присваивается весовой коэффициент значимости. Состав и содержание частных показателей определяются в соответствии с требованиями нормативно-правовых актов в области защиты ПДн. Так, например, для критерия ORG_1 «Подготовка персонала к безопасной обработке ПДн» частными показателями являются:

- определение круга лиц, допущенных к обработке ПДн;
- распределение обязанностей и персонификация ролей в управлении жизненным циклом ПДн и управлении инцидентами ИБ ПДн;
- ознакомление с положением по обеспечению безопасности ПДн при их обработке;
- ознакомление с мерами ответственности за нарушение правил защиты ПДн;
- установление персональной ответственности за нарушения правил обработки ПДн;
- реализация программ по обучению и повышению квалификации сотрудников при работе со средствами защиты информации.

Вычисление оценки по критерию ORG_1 «Подготовка персонала к безопасной обработке ПДн» выполняется по формуле (2.2):

$$ORG_i = \sum_{j=1}^{M} \alpha_{i,j}^{ORG} \times PF_{i,j}^{ORG}, \tag{2.2}$$

где $\alpha_{i,j}^{ORG}$ – весовой коэффициент j-ого частного показателя i-ого критерия организационного уровня ИБ ПДн; $PF_{i,j}^{ORG}$ – значение j-ого частного показателя i-ого критерия организационного уровня ИБ ПДн, вычисленное по правилам выставления баллов, описанным выше; M – общее количество критериев организационного уровня ИБ ПДн.

Аналогичным образом проводится оценка выполнения всех остальных критериев, а для оценки организационного уровня ИБ ПДн в целом необходимо определить среднее значение оценок критериев по формуле (3.2):

$$EV_{ORG} = \frac{\sum ORG_i}{M}, \tag{3.2}$$

где EV_{ORG} – оценка организационного уровня ИБ ПДн; ORG_i – значение оценки i-ого критерия организационного уровня ИБ ПДн.

Оценка программно-технического уровня ИБ ПДн

Оценка программно-технического уровня ИБ ПДн осуществляется с помощью критериев и частных показателей ИБ ПДн, позволяющих определить степень выполнения требований законодательства в области персональных данных (табл. 6).

Таблица 6

Набор критериев оценки
программно-технического уровня ИБ ПДн[4]

Обозначение критерия	Наименование критерия ИБ ПДн
PRT_1	Актуальность проектной и рабочей документации на СЗПДн
PRT_2	Выполнение требований безопасности для подсистемы управления доступом
PRT_3	Выполнение требований безопасности для подсистемы регистрации и учета
PRT_4	Выполнение требований безопасности для подсистемы обеспечения целостности
PRT_5	Выполнение требований безопасности для подсистемы контроля отсутствия недекларированных возможностей
PRT_6	Выполнение требований безопасности для подсистемы анализа защищенности
PRT_7	Выполнение требований безопасности для подсистемы обнаружения вторжений
PRT_8	Выполнение требований безопасности для подсистемы антивирусной защиты
PRT_9	Выполнение требований безопасности по межсетевому экранированию
PRT_{10}	Выполнение требований безопасности по закрытию технических каналов утечки ПДн
PRT_{11}	Выполнение требований безопасности по инженерно-технической защите помещений и средств вычислительной техники
PRT_{12}	Использование сертифицированных средств защиты

Для каждого критерия программно-технического уровня ИБ ПДн PRT_1–PRT_9 разрабатываются наборы частных показателей. Состав и содержание частных показателей определяются в соответствии с требованиями нормативно-правовых актов в области защиты ПДн. Так, например, для критерия PRT_1 «Актуальность проектной и рабочей документации на СЗПДн» частными показателями являются[5]:

– наличие пояснительной записки к техническому проекту;
– наличие описания комплекса технических средств;
– наличие описания программного обеспечения;
– наличие схем соединения внешних проводок;
– наличие чертежа установки технических средств;
– описание технологического процесса обработки данных;
– наличие программы и методики испытаний;
– наличие паспорта на СЗПДн.

Вычисление оценки по критерию PRT_1 «Актуальность проектной и рабочей документации на СЗПДн» выполняется по формуле (2.3):

$$PRT_i = \sum_{j=1}^{K} \alpha_{i,j}^{PRT} \times PF_{i,j}^{PRT}, \qquad (2.3)$$

где $\alpha_{i,j}^{PRT}$ – весовой коэффициент j-ого частного показателя i-ого критерия программно-технического уровня ИБ ПДн; $PF_{i,j}^{PRT}$ – значение j-ого частного показателя i-ого критерия программно-технического уровня ИБ ПДн, вычисленное по правилам выставления баллов, описанным выше; K – общее количество критериев программно-технического уровня ИБ ПДн.

Аналогичным образом проводится оценка выполнения всех остальных критериев, а для оценки программно-технического уровня ИБ ПДн в целом необходимо определить среднее значение оценок критериев по формуле (3.3):

$$EV_{PRT} = \frac{\sum PRT_i}{K}, \qquad (3.3)$$

где EV_{PRT} – оценка программно-технического уровня ИБ ПДн; PRT_i – значение оценки i-ого критерия программно-технического уровня ИБ ПДн.

Определение итогового уровня соответствия ИБ операторов ПДн требованиям законодательства в области защиты ПДн

Конечным результатом, который позволяет рассчитать предлагаемая методика, является качественная и количественная оценка итогового уровня соответствия ИБ операторов ПДн требованиям законодательства в области защиты ПДн.

Следуя делению шкалы оценок от 0 до 1 по принципу дихотомии, образуем 5 рангов со следующими диапазонами:

ранг 5 – (0,95; 1];
ранг 4 – (0,88; 0,95];
ранг 3 – (0,75; 0,88];
ранг 2 – (0,5; 0,75];
ранг 1 – [0; 0,5].

Каждая полученная оценка EV_{MNG}, EV_{ORG}, EV_{PRT} ранжируется в соответствии с ее значением и диапазонами шкалы.

Итоговый уровень соответствия определяется по наименьшему значению из трех оценок по направлениям оценки:

– оценки менеджмента ИБ ПДн организации;
– оценки организационного уровня ИБ ПДн организации;
– оценки программно-технического уровня ИБ ПДн организации.

Полученное значение итогового уровня ИБ операторов является основой при формировании заключения по результатам оценки выполнения требований законодательства в области защиты ПДн.

Примечания

1 *Галатенко В.А.* Категорирование информации и информационных систем. Обеспечение базового уровня информационной безопасности // Jet Info. 2006. № 4 (155). С. 2–26.

2 Стандарт Банка России «Обеспечение информационной безопасности организаций банковской системы Российской Федерации» СТО БР ИББС 1.2–2009. С. 7–9.

3 См.: Рекомендации по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных, ФСТЭК России от 15 февраля 2008 г. Типовые требования по организации и обеспечению функционирования шифровальных (криптографических) средств, предназначенных для защиты информации, не содержащей сведений, составляющих государственную тайну, в случае их использования для обеспечения безопасности персо-

нальных данных при их обработке в информационных системах персональных данных, 8 Центр ФСБ России от 21 февраля 2008 г.

4 Методические рекомендации по обеспечению с помощью криптосредств безопасности персональных данных при их обработке в информационных системах персональных данных с использованием средств автоматизации, 8 Центр ФСБ России от 21 февраля 2008 г. Основные мероприятия по организации и техническому обеспечению безопасности персональных данных, обрабатываемых в информационных системах персональных данных, ФСТЭК России от 15 февраля 2008 г.

5 См.: ГОСТ 34.201–89: Виды, комплектность и обозначение документов при создании автоматизированных систем. М., 1989.

И.В. Гайнанова

ПРИМЕНЕНИЕ ТЕОРИИ ИГР В АНАЛИЗЕ СКРЫТЫХ КАНАЛОВ С АКТИВНЫМ ПРОТИВНИКОМ

Целью данной работы является исследование модели поведения активного противника при условии, что он не анализирует поступающие в канал передачи данных сообщения, а только задерживает некоторые из них с определенной вероятностью q. С использованием аппарата теории игр найдены оптимальные вероятностные значения для различных моделей поведения агента и контролера, а также получена оценка пропускной способности скрытого канала в заданных условиях оптимальности.

Ключевые слова: скрытый канал, защита информации, активный противник, теория игр.

Модель скрытого канала лучше всего проиллюстрирована в так называемой «проблеме заключенных»[1]. Алиса (она выступает в роли агента) и Боб находятся в тюрьме в отдельных камерах далеко друг от друга и хотят организовать побег. Им можно общаться между собой путем передачи сообщений, которые не должны содержать никакого намека на побег. Вся их коммуникация наблюдается охранником (контролером) Ивом, который нарушит их планы и отправит в тюрьму более строгого режима, как только обнаружит хоть какой-то признак скрытого сообщения. Алисе и Бобу удастся реализовать свой план, если Боб получит сообщение со скрытой информацией от Алисы и Ив ничего не заподозрит.

С точки зрения защиты информации можно выделить две модели поведения контролера: пассивный (может только наблюдать за передачей данных) и активный (может модифицировать и даже удалять какие-то сообщения из потока). Первый случай имеет место, когда наличие скрытой информации или другого вида аномалий в канале передачи данных стоит под вопросом. Здесь основная

задача контролера – обнаружить факт передачи нелегитимированной информации, если такой произошел. Второй случай более распространен в открытых публичных сетях, где практически всегда существуют каналы утечки информации. Главная задача контролера – уменьшить пропускную способность таких каналов.

Таким образом, если Ив не меняет отправляемые Алисой сообщения, а только анализирует их на предмет наличия информации о плане побега, то его модель поведения соответствует пассивному состоянию. Если же Ив начинает модифицировать сообщения или даже задерживать их у себя, то его поведение уже соответствует активному состоянию. Далее при анализе активной модели поведения котролера, не ограничивая общности рассуждений, будем считать, что Ив задерживает сообщения.

Модель с пассивным контролером хорошо представлена в работе[2]. Показано, что если вероятностное распределение обычных сообщений совпадает с вероятностным распределением скрытых сообщений, то контролер не имеет возможности отличить скрытую передачу данных от обычной. Даже если распределения различны, но отличие минимально, то шансы контролера обнаружить факт наличия скрытой информации стремятся к нулю. Таким образом, возникает задача ограничения пропускной способности скрытого канала в случае, когда анализ входящей информации не позволяет сделать вывод о легитимности передаваемых сообщений. Для решения этой задачи используется модель активного противника.

В данной работе рассматривается модель, в которой контролер может задерживать часть сообщений в потоке, и при этом он никак не анализирует их на наличие скрытой информации. Учитывая, что агент будет неизменно подстраиваться под поведение контролера, в работе рассмотрена игровая модель скрытого канала и поставлена задача нахождения таких оптимальных вероятностных значений, при которых будет достигаться равновесие в системе.

Рассмотрим дискретный канал передачи данных, то есть будем считать, что за одну единицу времени по каналу проходит одно сообщение. Будем считать, что некоторые из приходящих сообщений содержат в себе скрытую информацию. В этом случае активному контролеру необходимо максимально снизить уровень несанкционированной передачи путем задержки части сообщений в канале. Выбор сообщений для задержки будет основываться на модели простой случайной выборки, то есть каждое сообщение в канале может быть задержано с некоторой фиксированной вероятностью $0 < q < 1$. Случай $q = 1$ не рассматривается, так как соответствует задержке каждого сообщения, то есть полной блокировке данного

канала передачи данных. Случай $q = 0$ не рассматривается, так как соответствует пассивной модели поведения контролера.

Очевидно предположить, что не каждое передаваемое в канале сообщение содержит скрытую информацию, так как в таком случае контролеру будет достаточно полностью заблокировать канал. Рассмотрим два случая поведения агента:

1) все сообщения отправляются Алисой независимо друг от друга;

2) существует зависимость между сообщениями, отправляемыми Алисой.

Случай независимых сообщений

Введем вероятность $0 < p < 1$, что очередное сообщение, отправленное Алисой, является скрытым. Найдем оценку на пропускную способность канала в данном случае.

Утверждение 1. В заданных предположениях минимальное число сообщений n, которое должно быть отправлено Алисой для гарантии, что с вероятностью $1-\alpha$ *(где* α *близко к 0) хотя бы одно скрытое сообщение дошло до Боба, удовлетворяет неравенству* $n \geq \log_{pq+1-p} \alpha$.

Доказательство.

$$P_1 = P(\text{хотя бы 1 скрытое сообщение дошло}) =$$

$$= 1 - P(\text{все скрытые сообщения задержаны}) = 1 - P_2 \geq 1 - \alpha \Rightarrow P_2 \leq \alpha$$

По формуле полной вероятности[3]

$$P_2 = \sum_{i=0}^{n} P(\text{все скрытые сообщения задержаны}|\text{скрытых сообщений i}) \cdot$$

$$\cdot P(\text{скрытых сообщений i}) = \sum_{i=0}^{n} P(\text{все i скрытых сообщений задержаны}).$$

Рассмотрим следующие случайные величины:

$$A_i = \begin{cases} 1 \text{ с вероятностью } p & \text{(в момент i отправлено скрытое сообщение)} \\ 0 \text{ с вероятностью } 1-p & \text{(в момент i отправлено обычное сообщение)} \end{cases}$$

$$W_i = \begin{cases} 1 \text{ с вероятностью } q & \text{(в момент i Ив задержал сообщение)} \\ 0 \text{ с вероятностью } 1-q & \text{(в момент i Ив пропустил сообщение)} \end{cases}$$

$$P(\text{все i скрытых сообщений задержаны}) =$$

$$= \sum_{k_1 < \ldots < k_i} P(A_{k_1} = \ldots = A_{k_i} = 1, A_{k_l} = 0 \text{ для } l = i+1, \ldots, n, W_{k_1} = \ldots = W_{k_i} = 1) =$$

$$= \{\text{все } A_i, W_i \text{ независимы}\} =$$

$$= C_n^i \left[P(A_i = 1)\right]^i \left[P(A_i = 0)\right]^{n-i} \left[P(W_i = 1)\right]^i = C_n^i p^i (1-p)^{n-i} q^i$$

$$P_2 = \sum_{i=0}^{n} C_n^i p^i (1-p)^{n-i} q^i =$$

$$= \left\{\text{по формуле бинома Ньютона } \sum_{i=0}^{n} C_n^i p^i q^{n-i} = (p+q)^n\right\} = (pq+1-p)^n \le \alpha$$

Отсюда $n \ge \log_{pq+1-p} \alpha$.

Найдем оптимальные значения p и q. Для этой цели рассмотрим игровую модель скрытого канала и будем считать оптимальными те значения, при которых достигается равновесие по Нэшу.

Имеется два игрока: агент Алиса A и контролер Ив W. Множество стратегий игрока A имеет вид $\{0,1\}$, где 0 соответсвует отправке обычного сообщения, а 1 – отправке скрытого сообщения. Аналогично для W множество стратегий имеет вид $\{0,1\}$, где 0 соответствует пропуску сообщения, а 1 – задержке сообщения. Опишем все возможные ситуации в игре в каждый момент времени и определим выигрыши игроков в каждом случае.

1. $(0,0)$. A отправляет обычное сообщение, W его пропускает. Выигрыши обоих игроков равны 0.

2. $(1,0)$. A отправляет скрытое сообщение, W его пропускает. Выигрыш A равен 1, так как он достигает своей основной цели, а именно успешно отправляет скрытое сообщение, выигрыш W равен -1, так как он оказывается неспособным предотвратить передачу.

3. $(1,1)$. A отправляет скрытое сообщение, W его задерживает. Выигрыш A равен -1, так как агенту не удалось передать скрытую информацию, выигрыш W равен 1, так как он смог предотвратить несанкционированную передачу данных.

4. $(0,1)$. A отправляет обычное сообщение, W его задерживает. Выигрыш A равен 0, так как задержка обычного сообщения никак не отражается на передаче скрытой информации, а выигрыш W равен $-1/2$, так как, с одной стороны, факта скрытой передачи не было, а с другой – Ив нарушил санкционированную передачу информации.

Получаем матричную игру, где $A = \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix}$ $W = \begin{pmatrix} 0 & -\frac{1}{2} \\ -1 & 1 \end{pmatrix}$.

$A_{ij} = A(i, j)$ – выигрыш Алисы в ситуации (i, j), $i \in \{0,1\}, j \in \{0,1\}$. Аналогично для Ива.

Известно, что в такой матричной игре не существует равновесия по Нэшу в чистых стратегиях[4]. Также по основной теореме матричных игр[5] известно, что в любой матричной игре существует равновесие в смешанных стратегиях. Для определения смешанных стратегий введем соответствующее вероятностное распределение на множестве стратегий каждого игрока в соответствии с ранее рассмотренными вероятностями p и q. Получаем для A $\{1-p, p\}$ и для $W\{1-q, q\}$.

Утверждение 2. В заданной игре оптимальными значениями p и q будут $\frac{1}{5}$ *и* $\frac{1}{2}$. *Выигрыши игроков будут равны 0* и $-\frac{1}{5}$ *соответственно.*

Доказательство. Выразим значения функций выигрыша через p и q.

$$A(p,q) = \sum_{i=0}^{1}\sum_{j=0}^{1} p_i A_{ij} q_j = (1-p)(A_{00}(1-q) + A_{01}q) + p(A_{10}(1-q) +$$

$$+A_{11}q) = p(1-2q)$$

$$W(p,q) = \sum_{i=0}^{1}\sum_{j=0}^{1} p_i W_{ij} q_j = (1-p)(W_{00}(1-q) + W_{01}q) + p(W_{10}(1-q) +$$

$$+W_{11}q) = 2.5pq - 0.5q - p$$

Заметим, что $\begin{array}{l}(W_{11} - W_{10})(W_{00} - W_{01}) = (1-(-1))(0-(-0.5)) = 1 > 0\\ (A_{11} - A_{01})(A_{00} - A_{10}) = (-1-0)(0-1) = 1 > 0\end{array}$

В этом случае оптимальные значения вероятностей задаются следующими равенствами[6]:

$$1 - p = \frac{W_{11} - W_{10}}{W_{11} - W_{10} + W_{00} - W_{01}}, \quad 1 - q = \frac{A_{11} - A_{01}}{A_{11} - A_{01} + A_{00} - A_{10}}$$

Подставляя соответствующие значения, находим оптимальные $p = \frac{1}{5}, q = \frac{1}{2}$. Из найденных ранее выражений для функций выигрыша получаем, что их значения равны 0 и $-\frac{1}{5}$ соответственно.

Используя *утв. 1* и *утв. 2,* получаем, что агенту A в заданных условиях необходимо отправить как минимум 22 сообщения, чтобы быть на 90 % уверенным, что по крайней мере одно скрытое сообщение было получено Бобом.

Случай зависимых сообщений

В реальности сообщения, которые отправлены абсолютно независимо друг от друга, наблюдаются очень редко. В случае скрытой передачи информации это связано также с тем, что агент A пытается варьировать отправку сообщений таким образом, чтобы максимально скрыть нелегитимированную передачу. В данном разделе будем считать, что процесс отправки сообщений A можно представить с помощью цепи Маркова.

Рассмотрим случайные величины A_i, определенные в предыдущем разделе. Будем считать, что легитимность первого отправленного сообщения (соответствует A_0) определяется аналогичным образом (0 с вероятностью $1-p$ и 1 с вероятностью p), а значения остальных определяются с помощью условных вероятностей.

$P(A_i = 1 \mid A_{i-1} = 1) = p_1$ – два скрытых сообщения подряд;

$P(A_i = 0 \mid A_{i-1} = 1) = 1 - p_1$ – после скрытого сообщения отправляется обычное;

$P(A_i = 1 \mid A_{i-1} = 0) = p_2$ – после обычного сообщения отправляется скрытое;

$P(A_i = 0 \mid A_{i-1} = 0) = 1 - p_2$ – два обычных сообщения подряд.

Утверждение 3. В заданных условиях оптимальные значения удовлетворяют следующим условиям: $p_1 + 4p_2 = 1, q = \frac{1}{2}$. *Выигрыши игроков будут 0 и* $-\frac{1}{5}$.

Доказательство. Определим вероятность того, что отправленное в момент n сообщение было скрытым $I_n = P(A_n = 1)$. По формуле полной вероятности[7]:

$$P(A_n = 1) = p_1 P(A_{n-1} = 1) + p_2 P(A_{n-1} = 0) = p_1 P(A_{n-1} = 1) +$$

$$+ p_2 \left[1 - P(A_{n-1} = 1)\right] = (p_1 - p_2) P(A_{n-1} = 1) + p_2 = I_{n-1}(p_1 - p_2) + p_2 =$$

$$= p_2 + p_2(p_1 - p_2) + I_{n-2}(p_1 - p_2)^2 = \ldots = I_0(p_1 - p_2)^n + p_2 \sum_{k=0}^{n-1} (p_1 - p_2)^k$$

По определению $I_0 = P(A_0 = 1) = p$, отсюда

$$P(A_n = 1) = p(p_1 - p_2)^n + p_2 \sum_{k=0}^{n-1} (p_1 - p_2)^k$$

Так как нас интересует вероятностное распределение в произвольный момент времени, то можно устремить n к ∞. Так как $p(p_1 - p_2)^n \to 0$ при $n \to \infty$ и $p_1, p_2 \in (0,1)$, то по формуле геометрической прогрессии

$$P(A_\infty = 1) = p_2 \sum_{k=0}^{\infty} (p_1 - p_2)^k = \frac{p_2}{1-(p_1 - p_2)}.$$

Посчитаем ожидаемое значение функции выигрыша для A и W.

$$E(A) = \sum_{i=0}^{1} \sum_{j=0}^{1} A(i, j) P(A_\infty = i, W_\infty = j) =$$

$$= \{\text{контролер и агент действуют независимо}\} =$$

$$= \sum_{i=0}^{1} \sum_{j=0}^{1} A(i, j) P(A_\infty = i) P(W_\infty = j) = (1 - 2q) \frac{p_2}{1-(p_1 - p_2)}$$

Аналогично $E(W) = -\frac{q}{2} + \left(\frac{5q}{2} - 1\right) \frac{p_2}{1-(p_1 - p_2)}$

Для нахождения равновесия по Нэшу воспользуемся тем же алгоритмом, что и в *утв. 2*, в качестве p здесь берется соответственно $\frac{p_2}{1-(p_1 - p_2)}$.

Получим, что оптимальные значения p_1, p_2 и q удовлетворяют условиям $q = \frac{1}{2}$ и $\frac{p_2}{1-(p_1 - p_2)} = \frac{1}{5}$, откуда $p_1 + 4p_2 = 1$ и по полученным ранее формулам

$$E(A) = 0 \text{ и } E(W) = -\frac{1}{4} + \left(\frac{5}{4} - 1\right) \frac{p_2}{1-(1 - 4p_2 - p_2)} = -\frac{1}{5}.$$

Из *утв. 3* следует неявное ограничение на p_2, а именно на отправку скрытых сообщений после обычных. Если вероятность такого события больше 1/4, то равновесие нарушается и большая часть сообщений задерживается контролером. Более того, указанная зависимость между p_1 и p_2 доказывает интуитивно понятный

факт, что при частой отправке скрытых сообщений они не должны быть оправлены подряд.

Кроме того, из *утв. 2* и *утв. 3* следует, что изменение агентом модели поведения не влияет на функцию выигрыша контролера при условии, что они оба действуют в соответствии с условиями оптимальности. Получаем, что выбранная стратегия контролера позволяет ему получить одинаковый выигрыш при различных моделях поведения агента.

Заключение

Полученные результаты имеют важное значение для предотвращения передачи больших объемов скрытой информации. Найденные значения позволяют оптимизировать действия контролера, что в том числе приведет к большей продуктивности его действий. В работе был рассмотрен случай, при котором контролер не анализирует входящие сообщения, таким образом, действия контролера и агента были независимы друг от друга. В реальных сетях это не всегда так, и большой интерес для дальнейших исследований представляет изучение смешанной модели поведения контролера, при которой он вначале анализирует сообщение, а затем уже принимает решение о его задержке. В этом случае вероятности p и q следует рассматривать не как стационарные, а как меняющиеся с течением времени. Кроме этого, процесс отправки сообщений агентом A может иметь более сложную структуру, чем модели, рассмотренные в данной работе. Исследование в этой области представляет собой огромный интерес, но из-за существенного усложнения модели требует проведения дополнительного анализа.

Примечания

1 *Simmons G.J.* The prisoners problem and the subliminal channel // Advances in Cryptology: Proceedings of Crypto 83 / Ed. by D. Chaum. Plenum Press, 1984. P. 51–67.

2 *Cachin C.* An Information-Theoretic Model for Steganography // Proc. 2nd Information Hiding Workshop. 1998. Vol. 1525. P. 306–318.

3 См.: *Ширяев А.Н.* Вероятность. М.: МЦНМО, 2004.

4 *Васин А.А., Морозов В.В.* Теория игр и модели математической экономики. М.: МАКС Пресс, 2005. С. 85–86.

5 Там же. С. 20.

6 Там же. С. 99–100.

7 См.: *Ширяев А.Н.* Указ. соч.

Е.И. Познякова

АНАЛИЗ ЭКСПЛУАТАЦИОННЫХ ПОТРЕБНОСТЕЙ С ЦЕЛЬЮ ОПРЕДЕЛЕНИЯ ТРЕБОВАНИЙ К КАЧЕСТВУ ОБСЛУЖИВАНИЯ

Методика вероятностного анализа эксплуатационных потребностей применялась до настоящего времени только для определения требований к качеству обслуживания железнодорожного транспорта. В данной статье рассматривается транспарентность методики для решения других задач, в частности для оценки показателей непрерывности бизнеса.

Ключевые слова: качество обслуживания, требования по безопасности, непрерывность бизнеса, функциональная безопасность.

В настоящее время многие организации используют в повседневной деятельности сложные новейшие технологии, но по данным последних исследований 80% всех компаний не способны восстановить свои бизнес-процессы после серьезного операционного сбоя. В связи с этим проблемы обеспечения непрерывности бизнеса являются актуальными в мире уже на протяжении десятков лет. Наиболее активно проявляют свой интерес к решениям в области непрерывности бизнеса финансовые, страховые компании и другие организации, чей бизнес в наибольшей степени зависит от информации. В РФ требование о наличии системы обеспечения непрерывности бизнеса определено разделом 8.11 стандарта Банка России «Обеспечение информационной безопасности организаций банковской системы РФ». Понятие функциональной безопасности (safety) как части общей безопасности (т. е. отсутствия недопустимого риска) для систем, которые зависят от правильности функционирования системы и оборудования, тесно связано с непрерывностью бизнеса. В частности, используемые в обоих областях показатели для оценки требований по существу аналогичны. Стандарты по функциональной безопасности, в том числе приня-

тый в РФ МЭК 61508[1], применяются при разработке высокотехнологичных критических систем, например для управления движением поездов на железных дорогах. С точки зрения работы с клиентами выделяются требования к качеству обслуживания и согласуется Соглашение об уровне услуг (SLA – Service Level Agreement), поскольку невозможно достижение 100% надежности информационных систем.

В современных исследованиях все три области (непрерывность бизнеса, функциональная безопасность, качество обслуживания) рассматриваются обособленно, однако представляется целесообразным использование общих методик для оценки аналогичных показателей и определения взаимосвязи требований, предъявляемых при разработке информационных систем. В настоящее время большая часть оценок основных показателей основана на экспертных и эмпирических данных. В данной статье рассмотрены методы вероятностного анализа, который позволяет определить критические значения с учетом эксплуатационных потребностей и требований к качеству обслуживания.

Идеология

Документ «ETCS/GSM-R Quality of Service – Operational Analysis»[2] был разработан рабочей группой EEIG ERTMS[3] Users Group с целью определения параметров качества обслуживания (QoS[4]) ETCS[5] с точки зрения эксплуатационных требований Европейской магистральной железнодорожной сети. Документ рассматривается как основа для определения технических требований QoS для поставщиков ETCS и GSM-R[6] и содержит методику операционного анализа для оценки соответствующих параметров. Рассмотрим представленную для железных дорог методику для общего случая – информационных систем любого профиля.

Пусть информационная система обеспечивает выполнение ряда функций для обеспечения производственного процесса на некотором предприятии. Рассмотрим понятие качества обслуживания относительно:

– предприятия в целом;
– информационной системы;
– компонентов информационной системы.

Все три составляющие связаны между собой. Характеристики компонентов влияют на общее качество обслуживания информационной системы. Поскольку мы будем рассматривать только предприятия, работа которых существенно зависит от функционирова-

ния информационной системы, то QoS предприятия напрямую связано с качеством информационной системы.

Подход к определению требований QoS включает:

– определение требований предприятия в целом с учетом его целей;
– анализ влияния основных сценариев работы и распределение требований верхнего уровня по этим сценариям;
– определение требований QoS для компонентов на основе анализа сценариев.

Реальная система никогда не достигает 100% производительности. Некоторый процент операций происходит со сбоем. В связи с этим возникают задержки в предоставлении услуг (выполнении сценариев или операций). Для анализа количества и длительности задержек построим гистограмму (рис. 1). По оси ординат – степень критичности, которая может выражаться в количестве выполненных операций. По оси абсцисс – время задержки при выполнении операции. Соответственно, если речь идет об информационной системе некоторого предприятия, это может быть время, которое потребовалось на восстановление бизнеса. Гистограмма отражает статистические данные, полученные в результате наблюдения за работой предприятия.

На основании гистограммы можно выделить границу, которая определяет процент операций, проводимых с критичной для предприятия задержкой.

Описанная идеология транспарентна и может быть применена для решения различных задач, в частности для оценки показателей обеспечения непрерывности бизнеса.

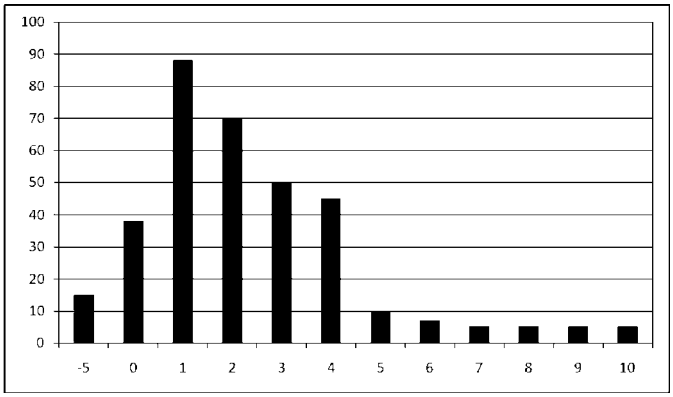


Рис. 1. Распределение количества операций, выполненных с задержкой

Расчет основных показателей

Обозначенные ниже критерии применяются для всего производственного процесса, но только их часть относится непосредственно к информационной системе. Для последующего исследования используется метод нисходящего анализа показателей для системы. Необходимо отметить, что результат определяется только экспертной оценкой на основе эмпирических данных о функционировании подобных систем.

Рассмотрим распределение в случае, когда задержка более N^7 минут допустима для 5% операций.

По статистическим исследованиям определяется соотношение задержек, вызванных техническими проблемами, человеческими факторами и условиями окружающей среды.

Часть всех используемых предприятием информационных технологий приходится на основную информационную систему. Задержка более N минут обычно происходит из-за сбоя оборудования, главным образом в связи с проблемами надежности. По этой причине весовой коэффициент, применяемый для оценки влияния QoS информационной системы, устанавливается равным 20%. Оставшиеся 80% задержек вызваны влиянием надежности оборудования. Следует отметить, что подобное распределение основано на экспертных оценках и определяется по результатам анализа деятельности предприятия.

С другой стороны, задержки в интервале от 0 до N минут наиболее вероятно происходят при влиянии QoS информационной си-

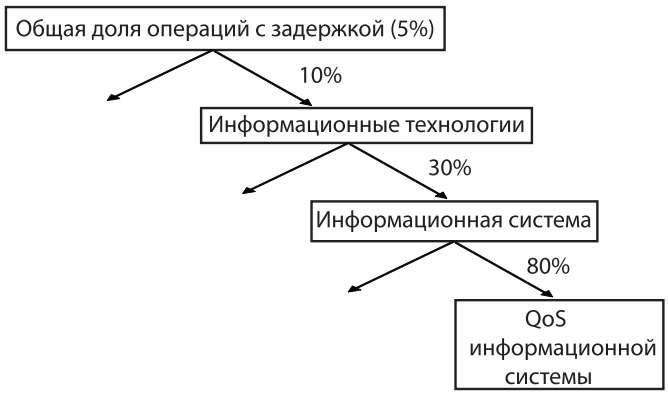


Рис. 2. Пример распределения при задержках более N минут

стемы. Весовой коэффициент для таких задержек равен 80% от более высокого уровня (уровня ETCS), описанного выше.

Методика, определенная в документе «ETCS/GSM-R Quality of Service» не учитывает зависимости между выполняемыми операциями. Вместе с тем сбой одной операции может повлечь цепную реакцию и привести к критическим последствиям для бизнеса. Сама по себе такая ситуация не приводит к катастрофе, но влияет на качество обслуживания.

Возможность сбоя при выполнении операций определена простым соотношением:

$$Q_{ИС\ QoS} = t_d / MTBF,$$

где t_d = время выполнения операции (например, 5 часов), MTBF – среднее время между сбоями в работе этого поезда (Mean time between failures), $Q_{ИС\ QoS}$ – вероятность прерывания с учетом влияний QoS информационной системы.

Поскольку $Q_{ИС}$ известно, соответствующее MTBF может быть вычислено. $Q_{ИС}$ QoS определяется следующим образом:

$Q_{ИС\ QoS} = 0{,}05 \times 0{,}1 \times 0{,}3 \times 0{,}2 = 0{,}030\%$ операций с задержкой > N минут.

Таким образом, вычисляем MTBF:

MTBF = t_d / 0,0003 = 16 667 часов (проблема QoS при задержке > N минут).

Анализ основных сценариев для железных дорог

Нисходящий анализ, описанный выше, позволяет определить ряд требований к MTBF. Рассмотрим соответствующие показатели для основных сценариев, возникающих при работе системы управления движением поездов ETCS.

Целевые показатели QoS ETCS высшего уровня должны быть определены для конкретных сценариев (начало работы, продление маршрута (MA – Movement Authority), переход на уровень 2). Поэтому проведен анализ каждого сценария с точки зрения его влияния на потенциальное опоздание поезда (суммарную задержку в рамках поездки) и пропускную способность линии. Так, нечастый переход на уровень 2 (уровень управления) в меньшей степени

влияет на потенциальное опоздание, чем сценарий продления маршрута поезда.

Помимо частоты выполнения сценария, необходимо понимать его последствия с точки зрения технологических задержек. Таким образом, общая задержка от влияния QoS ETCS является функцией частоты сбоев сценария и косвенной задержки.

Высокоуровневые целевые показатели QoS для ETCS должны быть распределены по соответствующим сценариям с учетом влияния каждого сценария на общую потенциальную задержку. Помимо частоты выполнения сценария, необходимо провести анализ последствий с точки зрения результирующей технологической задержки. Косвенная задержка зависит от ряда факторов, наиболее значимыми из которых являются:

– задержка при передаче;
– характеристики замедления и ускорения поезда;
– скорость на линии.

Распределение высокоуровневых целевых показателей по сценариям является сложной задачей. Тем не менее оценки влияния каждого сценария, проведенные железными дорогами, привели к следующим результатам:

– продление МА – 90%;
– переход на уровень 2 – 5%;
– начало / возобновление работы – 5%.

Теперь становится возможным выявить требования для каждого сценария. Для высокоскоростных линий ранее был определен показатель MTBF >5 минут.

Для общего QoS MTBF = 10 000 часов.

При распределении:

– продление МА 90% – MTBF = 10 000/0,9 = 11 111 ч;
– переход на уровень 25% – MTBF = 10 000/0,05 = 200 000 ч;
– начало / возобновление работы 5% – MTBF = 10 000/0,05 = = 200 000 ч.

После этого необходимо определить вероятность задержки в 5 минут для каждого сценария. Прежде всего нужно вычислить задержку ответной реакции системы для последующей технологической задержки в 5 минут. Анализ приведен в приложении D к документу[8]. Далее проводится оценка числа инцидентов для каждого сценария в час:

– продление МА – 100 / ч;
– переход на уровень 2–2 / ч[8];
– начало / возобновление работы – (1 / время поездки) × (1 / ч).

Допустимая вероятность сбоя работы определяется следующим образом.

Вероятность сбоя работы = 1/(MTBF сценария × частота сценария).

Для продления МА:

вероятность сбоя работы >5 минут = 1/(11 111 ч × 100 (1/г)) = = 9×10^{-7}.

Этот показатель QoS для сценария продления МА определен для скорости на линии 300 км/ч, т. е. одно из каждого 1,1 миллиона ($1/9 \times 10^{-7}$) обновлений МА может пройти с задержкой более 246 сек.

Непрерывность бизнеса

Описанная методика может быть применена при решении задач не только для железных дорог. В частности, очевидна значимость подобного анализа для оценки директивного времени восстановления (RTO – Recovery Time Objective) при обеспечении непрерывности бизнеса.

В этом случае можно определить параметры QoS для каждого уровня иерархии:

- для предприятия в целом – требования обеспечения непрерывности бизнеса;
- для информационной системы – значение времени восстановления (RTO);
- для компонентов – количественные требования для интенсивности отказов.

Значение времени восстановления определяется на основе статистических данных, после чего вычисляется вероятность сбоя, нарушающего непрерывность бизнеса.

Дальнейшие исследования должны быть направлены на анализ методов оценки показателей функциональной безопасности и выявление связи требований по ФБ и непрерывности бизнеса. Это позволит разработать комплексную методику определения требований к критическим технологиям.

Заключение

В настоящее время оценка основных показателей непрерывности бизнеса, например RTO, проводится каждой компанией самостоятельно, а основные проблемы и угрозы рассматриваются узко, только с точки зрения катастрофоустойчивости. Данное исследование позволит расширить спектр анализируемых угроз и произво-

дить более точные расчеты основных критериев. Применение методик анализа требований к функциональной безопасности и качеству обслуживания, апробированных на практике в других областях, позволит учесть эксплуатационные потребности и проблемы, возникающие при основных сценариях работы.

Примечания

1 См.: ГОСТ Р МЭК 61508-2007. Функциональная безопасность систем электрических, электронных, программируемых электронных, связанных с безопасностью. М.: Стандартинформ, 2008.

2 См.: EEIG: 04E117 (ETCS/GSM-R Quality of Service – Operational Analysis – Качество обслуживания ETCS/GSM-R – Операционный анализ). ERTMS Users group, 2005.

3 ERTMS (European Railway Traffic Management System) – Европейская система управления железнодорожным движением.

4 QoS (Quality of Service) – качество обслуживания.

5 ETCS (European Train Control System) – Европейская система управления поездом.

6 GSM-R (Global System for Mobiles – Railway) – Глобальная система связи для железных дорог.

7 Временна́я граница и доля операций с задержкой определяются в соответствии с гистограммой.

8 См. EEIG: 04E117.

Д.К. Скачек

ПОСТРОЕНИЕ СОСТОЯТЕЛЬНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ КРИТЕРИЕВ ПРИ УСЛОВИИ НЕСОГЛАСОВАННОСТИ МЕР

Существование состоятельной последовательности критериев говорит о возможности выявления аномального поведения системы. Предполагаем, что состояние системы описывается конечномерным топологическим пространством, на каждом из элементов которого задана вероятностная мера. Цель данной работы – построить пример семейства вероятностных мер, не удовлетворяющих условиям согласованности, и таких, что можно построить состоятельную последовательность критериев для определения аномального поведения.

Ключевые слова: несогласованные меры, состоятельная последовательность критериев, проверка гипотез.

Вопрос о существовании состоятельной последовательности критериев возникает в задачах, описываемых конечными дискретными моделями, например в задачах, связанных с выявлением изменения параметров, определяющих поведение системы. Допустим, что нормальное поведение системы характеризуется некоторым вероятностным распределением. Отклонение поведения системы (аномальное поведение) описывается семейством распределений, но при других параметрах. В этом случае возникает задача выявления аномального поведения. Однако, чтобы делать какие-либо выводы из результатов наблюдения за поведением системы, необходимо знать возможности статистического инструментария по выявлению отклонений. Таким образом, возникает вопрос о существовании статистической процедуры, позволяющей обнаружить аномальное поведение. Существование состоятельной последовательности критериев характеризует возможность выявления отклонения системы от нормального поведения. С другой

стороны, их несуществование говорит о том, что может быть построен статистически невыявляемый скрытый канал.

Пусть $X = \{x_1, \ldots, x_m\}$ – конечное множество, определяющее последовательность конечных множеств $X, X^2, \ldots, X^n, \ldots$, которые являются множествами последовательностей длины n, $n = 1, 2, \ldots$, с элементами из X. Считаем, что на каждом из множеств этой последовательности заданы вероятностные меры

$$P_{0,n}(x_1, \ldots x_n),\ x_i \in X,\ n = 1, 2, \ldots \quad (1)$$

$$P_{\theta,n}(x_1, \ldots x_n),\ x_i \in X,\ \theta \in \Theta,\ n = 1, 2, \ldots \quad (2)$$

Введем пространство бесконечных последовательностей $X^\infty = \{x_1, \ldots, x_n\},\ x_i \in X,\ n = 1, 2, \ldots$

Пусть поведение системы описывается некоторой случайной величиной ξ. Проверяется простая гипотеза $H_{0,n}: \xi \sim P_{0,n}$ против сложной альтернативы $H_{1,n}: \xi \sim P_{\theta,n},\ \theta \in \Theta$.

Для задания критерия проверки гипотез $H_{0,n}$ против $H_{1,n}$ при уровне значимости α_n необходимо определить критические множества $S_n \in \mathrm{A}\left(X^n\right)$ такие, что $P_{0,n}(S_n) < \alpha_n$. Мощность данного критерия – это функция $W_n(\theta) = P_{\theta,n}(S_n)$. Здесь $\mathrm{A}\left(X^n\right)$ – σ-алгебра всех цилиндрических подмножеств из X_n.

Пусть $T_n, n = 1, 2, \ldots$ – последовательность статистических критериев для проверки гипотез $H_{0,n}$ против альтернатив $H_{1,n}$ с критическими множествами S_n. Последовательность $T_n, n = 1, 2, \ldots$ называется состоятельной, если при $n \to \infty$ последовательность уровней значимости $\alpha_n \to 0$, а последовательность мощностей $W_n(\theta) \to 1$ для каждого параметра $\theta \in \Theta$[1].

В случае, когда вероятностные меры (1), (2) удовлетворяют условиям согласованности[2], по теореме Колмогорова вероятностные меры (1) определяют единственную меру P_0 на измеримом пространстве $\left(X^\infty, \mathrm{A}\right)$, и для каждого $\theta \in \Theta$ семейство конечномерных распределений (2) определяет единственную меру P_θ на измери-

мом пространстве $\left(X^{\infty}, \mathrm{A}\right)$. На основе свойств некоторых множеств в тихоновских произведениях в бесконечномерном пространстве в работе Грушо и Тимониной[3] доказаны теоремы о существовании состоятельных последовательностей критериев $T_n, n = 1, 2, \ldots$ для проверки гипотез $H_{0,n}$ против альтернатив $H_{1,n}$ при условии существования открытого покрытия носителя меры P_0.

Кроме того, в случае существования замкнутых покрытий носителей мер P_0 и P_θ определены условия, при которых возможно принятие решения на конечном шаге n. Несуществование состоятельной последовательности критериев в конечных вероятностных схемах является следствием выполнения некоторых простых достаточных условий[4], которым должны удовлетворять продолжения вероятностных мер в бесконечное произведение, порожденное исходным конечным пространством.

Приведенные результаты были получены для случая согласованных вероятностных мер (1), (2). В данной работе определяется семейство вероятностных мер, которое не является согласованным. В этом случае без каких-либо предположений топологического характера о структуре рассматриваемых измеримых пространств или о структуре семейства мер $P_{\bullet,n}$ теорема Колмогорова о продолжении мер в бесконечномерное измеримое пространство может быть неверна[5]. То есть нельзя утверждать, что существует вероятностная мера на измеримом пространстве $\left(X^{\infty}, \mathrm{A}\right)$, где A – σ-алгебра, порожденная всеми цилиндрическими множествами из X^{∞}.

Пример

Рассмотрим простую вероятностную модель передачи последовательности сообщений от компьютерной системы КА к компьютерной системе КВ через некоторый канал связи S. Будем считать, что поток сообщений однонаправленный. Кроме того, представим, что противник КА1 действует независимо в компьютерной системе КА и посылает скрытые сообщения через тот же канал связи своему напарнику КВ1, который находится в системе КВ. Сообщения принадлежат некоторому конечному множеству, а их выбор для передачи подчиняется сложному вероятностному закону, который мы не рассматриваем.

Пусть интервалы между сообщениями принимают значения 0 или 1. Тогда после получения $(n+1)$-го сообщения мы имеем последовательность $(\omega_1,\ldots,\omega_n)$ длины n, состоящую из нулей и единиц. Предположим, что нормальное поведение системы характеризуется ситуацией, при которой количество единиц в рассматриваемой последовательности равно $\left[\frac{n}{2}\right]$, где $[\bullet]$ – целая часть числа. Построим вероятностную меру $P_{0,n}$, заданную на измеримом пространстве $\left(X^n, \mathrm{A}(X^n)\right)$, где X^n – множество последовательностей длины n с элементами {0, 1}:

$$P_{0,n}(\omega_1,\ldots,\omega_n)=\begin{cases}\frac{1}{M}, \sum\limits_{i=1}^{n}\omega_i=\left[\frac{n}{2}\right];\\ 0.\end{cases}$$

Здесь M – количество последовательностей длины n, в которых количество единиц равно $\left[\frac{n}{2}\right]$, то есть $M=C_n^{[n/2]}$.

Покажем теперь, что это семейство мер является несогласованным. Пусть $n = 2k$, тогда $M=C_{2k}^{k}$. При $n = 2k +1$ получаем $M=C_{2k+1}^{k}$. Очевидно, что для любого номера n и любой последовательности из нулей и единиц длины n

$$\frac{(2k)!}{k!\cdot k!}\neq\frac{(2k+1)!}{k!\cdot(k+1)!},$$

то есть условие согласованности не выполняется.

Предположим, что аномальное поведение описывается семейством вероятностных мер

$$P_{\theta,n}(\omega_1,\ldots,\omega_n)=\begin{cases}\frac{1}{M_\theta}, \sum\limits_{i=1}^{n}\omega_i=\theta,\ \theta\in\Theta;\\ 0.\end{cases}$$

Здесь множество $\Theta=0,1,\ldots,\left[\frac{n}{2}\right]-1,\left[\frac{n}{2}\right]+1,\ldots,n$. Это семейство вероятностных мер также является несогласованным для каждого $\theta\in\Theta$. Проверяются гипотезы $H_{0,n}: P_{0,n}$ против альтернатив $H_{1,n}: P_{\theta,n},\ \theta\in\Theta$.

В построенном примере носители мер $P_{0,n}$ и $P_{\theta,n}$ не пересекаются, значит, существует состоятельная последовательность критериев для проверки гипотезы $H_{0,n}: P_{0,n}$ против альтернатив $H_{1,n}: P_{\theta,n},\ \theta \in \Theta$[6].

Заключение

Рассматривая задачу о существовании состоятельной последовательности критериев в последовательностях конечных вероятностных схем, мы отказались от согласованности мер, заданных на каждом из рассматриваемых n-мерных пространств, что привело нас к невыполнению условий существования вероятностной меры в пространстве бесконечных последовательностей. Была построена вероятностная модель с несогласованными мерами и показано, что при определенных условиях состоятельная последовательность критериев для выявления аномального поведения системы существует.

Примечания

1 *Грушо А.А., Тимонина Е.Е.* Некоторые связи между дискретными статистическими задачами и свойствами вероятностных мер на топологических пространствах // Дискретная математика. М., 2006. Т. 18. Вып. 4. С. 128–135.

2 См.: *Ширяев А.Н.* Вероятность. М.: МЦНМО, 2004.

3 См.: *Грушо А.А., Тимонина Е.Е.* Указ. соч.

4 *Грушо А.А., Грушо Н.А., Тимонина Е.Е.* Теоремы о несуществовании состоятельных последовательностей критериев в некоторых дискретных задачах // Дискретная математика. М., 2008. Т. 20. Вып. 2. С. 26–31; *Grusho A., Grebnev N., Timonina E.* Covert channels invisibility theorem // MMM-ACNS 2007, CCIS 1. St. Petersburg, 2007. Springer. P. 187–196.

5 См.: *Ширяев А.Н.* Указ. соч.

6 См.: *Грушо А.А., Тимонина Е.Е.* Указ. соч.

А.В. Гусев

МЕТОДЫ СЛУЧАЙНЫХ ГРАФОВ ДЛЯ СТЕГОАНАЛИЗА КОНТЕЙНЕРОВ, ПРЕДСТАВИМЫХ В ВИДЕ ГАУССОВСКИХ ПРОЦЕССОВ

В данной работе с помощью метода компьютерного моделирования исследуется работоспособность стеганографической схемы, построенной на гауссовских процессах. Исследуется возможность обнаружения в контейнере стеганографической вставки методом случайных графов.

Ключевые слова: стеганография, компьютерное моделирование, гауссовский процесс, случайные графы.

Метод формирования стеганографической вставки

В качестве стеганографического контейнера используется стационарный гауссовский процесс[1]. Каждому из символов алфавита, из которого формируется стеганографическая вставка, ставится в соответствие отрезок случайного процесса, аналогичного контейнеру и имеющего такое же распределение. Все такие отрезки процессов, сопоставляемых символам алфавита, имеют одинаковую длину (далее – отрезки). Отрезки, соответствующие символам встраиваемого сообщения, последовательно соединяются друг с другом и накладываются на контейнер с коэффициентами λ_1 (для контейнера) и λ_2 (для вставки). Коэффициенты выбираются таким образом, чтобы контейнер с наложенной вставкой имел такое же распределение, как и контейнер без вставки.

Метод извлечения стеганографической вставки

Получателю стеганографического контейнера известен алфавит стеганографической вставки и отрезки, соответствующие каждому из символов этого алфавита. Для извлечения стеганографической вставки из контейнера получатель разделяет контейнер на части, длины которых равны длине отрезков. Из каждой такой части он поочередно вычитает отрезок, сопоставленный каждому из символов алфавита (с коэффициентом λ_2), для полученной разности вычисляет эмпирическую дисперсию. Если для некоторого символа дисперсия разности близка к $\lambda_1^2\sigma^2$, где σ^2 – дисперсия контейнера, получатель делает вывод о том, что данная часть контейнера содержит этот символ.

Метод обнаружения

Анализируемый на наличие стеганографической вставки отрезок процесса разделяется на части, длины которых равны предполагаемой длине отрезков, сопоставленных алфавиту. Для каждой пары полученных отрезков формируется отрезок процесса, являющийся разностью анализируемой пары. Затем вычисляется эмпирическая дисперсия анализируемых отрезков и их разности. В случае, если оба анализируемых отрезка содержат стеганографическую вставку одного и того же символа, дисперсия разности будет равна $2\lambda_1^2\sigma^2$, в противном случае – $2\sigma^2$. Обнаружение в анализируемом процессе пар с одинаковыми символами означает наличие стеганографической вставки и выявляет повторяющиеся символы скрытого сообщения[2].

Описание модели

Для анализа выявления стегосхемы использовался метод компьютерного моделирования. В реализованной модели стеганографического канала в качестве процесса использовалась последовательность псевдослучайных чисел, которые имитировали стационарный гауссовский процесс с параметрами одномерного распределения (0,1). Псевдослучайные числа с нормальным распределением были получены путем преобразования значений, полученных из библиотечной функции генерации псевдослучайных чисел с равномерным распределением с помощью преобразования Бокса–Мюллера[3].

Для каждого из символов алфавита фиксированной длины формировалась последовательность псевдослучайных чисел длины n. Затем для каждого символа заданного заранее текста длины k в этом алфавите формировалась последовательность такой же длины, имитирующая контейнер. Две полученные последовательности поэлементно складывались с коэффициентами λ_1 и $\lambda_2 = \sqrt{1-\lambda_1^2}$ и формировали суммарную последовательность, имитирующую контейнер, содержащий стеганографическую вставку.

Контейнер, содержащий скрытое сообщение полностью, получается путем последовательного объединения полученных сумм и содержит $n \cdot k$ значений. Распределение контейнера со вставкой имеет те же параметры, что и контейнер без вставки.

После формирования контейнера со вставкой производится анализ наличия в нем вставок по изложенному выше методу. Контейнер разделяется на k подпоследовательностей длины n каждая. Для каждой пары полученных подпоследовательностей формируется последовательность, являющаяся поэлементной разницей пары подпоследовательностей, для сформированной последовательности вычисляется дисперсия и заносится в таблицу. Если подпоследовательности содержат одинаковые стеганографические вставки, значения эмпирической дисперсии будут близки к $2\lambda_1^2$, для остальных пар – к 2.

Далее по полученной таблице производится поиск значений, существенно меньших 2. Отсутствие таких значений означает отсутствие стеганографической вставки в контейнере, наличие таких значений означает наличие вставки; пары подпоследовательностей, для которых получены эти значения, содержат вставку одинакового скрытого символа.

Пример. Используется алфавит {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}; стеганографическая вставка (5, 6, 2, 0, 9, 4, 1, 4, 3, 7, 0, 9, 4, 6, 4, 3, 6, 7, 1, 5); длина подпоследовательностей контейнера n = 10000, контейнер длины $n \cdot k$, k = 20, значение λ_1 = 0,700. Вычислено значение $\lambda_2 \approx 0{,}714$.

Моделирование выделения стеганографической вставки из контейнера

Для выделения стеганографической вставки из контейнера использовался описанный выше метод. Значения оценок дисперсий, полученные при выделении стеганографической вставки приемни-

ком, приведены в таблице 1. Столбцы таблицы от 0 до 9 соответствуют символам алфавита, строки – частям контейнера, в которых производился поиск вставок. В ячейках таблицы содержатся значения дисперсий, полученные для разности части контейнера и последовательности, соответствующей символу алфавита вставок. Цветом выделены ячейки, в которых полученное значение дисперсии близко к $\lambda_1^2 = 0{,}49$. Последний столбец каждой строки содержит указание на то, какой символ был обнаружен в части контейнера, соответствующей строке. Текст стеганографической вставки, представляющий собой последовательность символов 5, 6, 2, 0, 9, 4, 1, 4, 3, 7, 0, 9, 4, 6, 4, 3, 6, 7, 1, 5, был извлечен верно.

Таблица 1

Извлечение стеганографической вставки

Отрезок №	Символы										Обнаруженный символ
	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"	
0	1,4968	1,5010	1,4846	1,4900	1,4846	**0,4784**	1,5147	1,5154	1,5090	1,4936	5
1	1,4681	1,4992	1,5116	1,5039	1,5044	1,5500	**0,4840**	1,5020	1,5382	1,4926	6
2	1,4782	1,5186	**0,4828**	1,5294	1,4887	1,4978	1,5129	1,5204	1,5194	1,5130	2
3	**0,4860**	1,5080	1,4992	1,5273	1,5047	1,5270	1,4790	1,4916	1,5236	1,4869	0
4	1,5061	1,5178	1,5229	1,5189	1,5280	1,5410	1,5207	1,5192	1,5096	**0,4840**	9
5	1,5111	1,5011	1,5148	1,4800	**0,4873**	1,5237	1,5150	1,5246	1,5225	1,5203	4
6	1,4965	**0,4819**	1,5216	1,5501	1,5235	1,5402	1,5300	1,5505	1,5666	1,5311	1
7	1,4900	1,5372	1,5077	1,4838	**0,4877**	1,5056	1,5280	1,5267	1,5161	1,5223	4
8	1,5171	1,5058	1,5442	**0,4848**	1,4770	1,5065	1,5193	1,5314	1,5275	1,5005	3
9	1,5075	1,5318	1,5328	1,5380	1,5268	1,5431	1,5364	**0,4801**	1,5507	1,5343	7
10	**0,5030**	1,5208	1,5217	1,5375	1,5128	1,5440	1,5142	1,5085	1,5364	1,5120	0
11	1,5098	1,5234	1,5281	1,4924	1,5102	1,5150	1,4756	1,5116	1,4980	**0,4760**	9
12	1,4816	1,4810	1,4778	1,4626	**0,4830**	1,4822	1,5015	1,5056	1,4906	1,5084	4
13	1,5014	1,5090	1,5161	1,5192	1,5128	1,5489	**0,4917**	1,5118	1,5251	1,5134	6
14	1,5142	1,5197	1,5235	1,4926	**0,4956**	1,5259	1,5233	1,5492	1,5305	1,5384	4

Окончание табл. 1

Отрезок №	Символы										Обнаруженный символ
	"0"	"1"	"2"	"3"	"4"	"5"	"6"	"7"	"8"	"9"	
15	1,5019	1,5099	1,5020	**0,4859**	1,4640	1,5102	1,5027	1,4929	1,5116	1,4925	3
16	1,5200	1,5614	1,5692	1,5398	1,5506	1,5544	**0,5015**	1,5276	1,5590	1,5369	6
17	1,5333	1,5460	1,5553	1,5570	1,5507	1,5703	1,5565	**0,5048**	1,5526	1,5679	7
18	1,5000	**0,4947**	1,5353	1,5320	1,5175	1,5302	1,5139	1,5266	1,5618	1,5396	1
19	1,5273	1,5301	1,5176	1,5160	1.5179	**0,4935**	1,5352	1,5432	1,5427	1,5397	5

Моделирование обнаружения стеганографической вставки

Для обнаружения стеганографической вставки в контейнере использовался описанный выше метод. Значения оценок дисперсий, полученные при поиске стеганографической вставки приемником, приведены в таблице 2. Номера столбцов и строк таблицы соответствуют порядковым номерам частей контейнера, которые сравнивались в процессе поиска. В ячейках таблицы указана полученная оценка дисперсии для данной пары. Цветом выделены ячейки, в которых полученное значение дисперсии близко к $2\lambda_1^2 = 0{,}98$. Были обнаружены совпадения символов с порядковыми номерами:

0 и 19 (символ «5»)

1, 13 и 16 (символ «6»)

3 и 10 (символ «0»)

4 и 11 (символ «9»)

5, 7, 12 и 14 (символ «4»)

6 и 18 (символ «1»)

8 и 15 (символ «3»)

9 и 17 (символ «7»).

Обнаруженные совпадения соответствуют тексту стеганографической вставки.

Таблица 2

Обнаружение стеганографической вставки

№ части Контей-нера																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	2,020																		
2	1,974	1,997																	
3	1,984	1,929	1,967																
4	2,003	1,997	2,011	1,965															
5	1,995	1,995	1,986	2,016	2,033														
6	2,001	2,000	1,999	1,993	2,011	2,019													
7	1,961	2,010	1,967	1,998	2,021	**0,983**	2,018												
8	1,985	1,988	2,044	2,015	1,992	1,963	2,037	1,971											
9	2,021	2,035	2,012	1,998	2,017	2,020	2,062	2,005	2,024										
10	2,001	1,978	1,984	**0,986**	2,002	2,007	2,013	2,017	2,029	2,016									
11	1,966	1,948	1,996	1,969	**0,977**	2,020	2,014	2,010	1,979	2,019	2,002								
12	1,939	1,997	1,950	1,980	2,040	**0,958**	1,980	**0,981**	1,963	2,003	1,982	2,008							
13	2,009	**0,976**	1,993	1,960	2,028	1,999	2,009	1,994	2,011	1,995	2,004	1,985	1,976						
14	1,970	1,998	1,998	2,019	2,063	**0,994**	2,016	**0,986**	1,990	2,039	2,004	2,026	**0,979**	2,009					
15	1,968	1,986	1,976	1,997	2,002	1,967	2,012	1,953	**0,981**	2,001	2,003	1,945	1,938	2,008	1,962				
16	2,001	**1,000**	2,068	2,004	2,009	2,019	2,044	2,049	2,033	2,027	2,040	1,990	2,027	**0,994**	2,064	2,014			
17	2,056	2,027	2,043	2,013	2,058	2,029	2,078	2,053	2,059	**0,981**	2,028	2,042	2,006	2,035	2,068	2,006	2,057		
18	1,993	2,000	2,013	1,986	2,021	1,998	**0,974**	2,039	2,009	2,037	1,992	2,038	1,987	1,980	2,012	2,005	2,052	2,043	
19	**0,994**	2,016	1,994	2,026	2,028	2,002	2,043	1,994	2,002	2,030	2,034	1,987	1,967	2,021	2,022	2,002	2,043	2,059	2,034

Выводы

Полученные результаты дают возможность сделать следующие выводы:

– описанная стеганографическая схема позволяет получателю стегоконтейнера извлекать стеганографическую вставку при условии знания отрезков процессов, сопоставленных алфавиту вставки;
– описанный метод выявления стеганографической вставки позволяет обнаруживать наличие вставки при условии, что 1) текст вставки содержит повторяющиеся символы и 2) известна или угадана длина отрезков процессов, сопоставленных алфавиту вставки.

Примечания

1 См.: *Прохоров Ю.В., Розанов Ю.А.* Теория вероятностей. М.: Наука, 1993.

2 См.: *Грушо А.А.* Некоторые статистические задачи на графах // Математические заметки. 1984. Т. 36. Вып. 2; *Buell D.A.* Calibrating Entropy Functions Applied to Computer Networks // Proceedings of MMM-ACNS. 2005. P. 76–87.

3 *Box G.E.P., Muller M.E.* A Note on the Generation of Random Normal Deviates // The Annals of Mathematical Statistics. 1958. Vol. 29. No. 2. P. 610–611.

С.Ю. Мельников

МНОГОУГОЛЬНИКИ, ХАРАКТЕРИЗУЮЩИЕ СТАТИСТИЧЕСКИЕ СВОЙСТВА БУЛЕВЫХ ФУНКЦИЙ В СХЕМЕ РЕГИСТРА СДВИГА

Булевой функции f от n аргументов ставится в соответствие геометрическое место точек, координаты которых есть пределы относительных частот встречаемости единиц во входной и выходной последовательностях двоичного проходного регистра сдвига с накопителем размера n и функцией выходов f. Это геометрическое место точек является выпуклым многоугольником. Исследуется, как алгебраические свойства функции f отражаются в геометрических свойствах ее многоугольника.

Ключевые слова: граф де Брейна, регистр сдвига, выпуклый многоугольник, булева функция.

Проблематике классификации k-значных и булевых функций по различным принципам посвящено значительное количество исследований[1]. Различные принципы классификации, как правило, связаны с различными автоматными схемами, в которых дискретная функция участвует в качестве функции выходов или переходов.

В настоящей работе мы рассматриваем булеву функцию как функцию выходов неавтономного двоичного проходного регистра сдвига. Совместные значковые статистические свойства входной и выходной последовательностей такого автомата можно характеризовать с помощью его многоугольника[2]. В работе изучается, как свойства булевой функции отражаются в геометрических свойствах такого многоугольника. Работа состоит из пяти параграфов. В первом дается определение, во втором указывается алгоритм построения многоугольников, основанный на перечислении циклов графа де Брейна, в третьем исследуются простейшие геометрические характеристики многоугольника, связанные с отношениями

двойственности и весом булевой функции. В четвертом параграфе строятся многоугольники некоторых линейных функций и функций, инвариантных относительно циклического сдвига переменных, в пятом описываются многоугольники функций малого числа переменных и приводятся примеры с результатами компьютерных экспериментов.

Определение многоугольника булевой функции

Пусть V_n – множество всех n-мерных двоичных векторов, F_n – множество всех булевых функций от n аргументов, $n = 1, 2, \ldots$. Для булевой функции $f(x_1, x_2, \ldots, x_n) \in F_n$ через $A_f = (X = \{0,1\}, V_n, Y = \{0,1\}, h, f)$ обозначим автомат Мура, являющийся двоичным регистром сдвига с накопителем размера n, множеством состояний V_n, функцией переходов h, определяемой по правилу $h((\alpha_1, \ldots, \alpha_n), x) = (\alpha_2, \ldots, \alpha_n, x)$, где $x, \alpha_i \in \{0,1\}$, $i = 1, 2, \ldots, n$.

Графом переходов такого автомата является двоичный граф G_n де Брейна степени n, т. е. ориентированный граф с множеством вершин V_n, содержащий дугу, выходящую из вершины $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ и заходящую в вершину $(\beta_1, \beta_2, \ldots, \beta_n)$ в том и только в том случае, когда $(\alpha_2, \alpha_3, \ldots, \alpha_n) = (\beta_1, \beta_2, \ldots, \beta_{n-1})$. Будем считать, что такая дуга помечена «входным» символом β_n и «выходным» символом $f(\alpha_1, \alpha_2, \ldots, \alpha_n)$. Через $C(G_n)$ обозначим множество всех простых циклов в G_n.

Предположим, что автомат A_f, начиная работать из некоторого начального состояния, перерабатывает двоичную последовательность $\chi^{(N)} = \left(x^{(0)}, x^{(1)}, \ldots, x^{(N-1)}\right)$ в двоичную последовательность $\gamma^{(N)} = \left(y^{(0)}, y^{(1)}, \ldots, y^{(N-1)}\right)$, $N \geq 1$. Пусть $p^{(N)}$ и $\pi^{(N)}$ – относительные частоты встречаемости единицы в последовательностях $\chi^{(N)}$ и $\gamma^{(N)}$ соответственно, $p^{(N)} = \frac{1}{N}\sum_{i=0}^{N-1} x^{(i)}$, $\pi^{(N)} = \frac{1}{N}\sum_{i=0}^{N-1} y^{(i)}$.

Через T обозначим множество всех периодических последовательностей (возможно, с подходом) в алфавите $\{0,1\}$.

Последовательности $\chi \in \mathrm{T}$ с началом $\chi^{(N)}$ поставим в соответствие вектор

$$z_{A_f}(\chi) = \lim_{N\to\infty}\left(p^{(N)}, \pi^{(N)}\right).$$

Предел справа существует в силу периодичности входной и выходной последовательностей и не зависит от начального состояния автомата в силу его сильной связности.

Для автомата A_f это правило определяет отображение

$$Z_{A_f} : \mathrm{T} \to [0,1]\times[0,1].$$

Основным предметом исследования настоящей работы является замыкание (совокупность всех предельных точек) множества $Z_{A_f}(\mathrm{T})$, которое мы будем обозначать R_f и называть многоугольником булевой функции *f*,

$$R_f = \left[Z_{A_f}(\mathrm{T})\right]. \qquad (1)$$

В работе показано, что такое определение можно распространить на более широкий класс входных последовательностей (чезаровские последовательности)[3].

Из результатов следуют две теоремы[4].

<u>Теорема 1.</u> R_f *– выпуклый многоугольник в квадрате* $[0,1]\times[0,1]$, $R_f = Conv\left\{\left(\frac{\nu(c)}{l(c)}, \frac{\mu(c)}{l(c)}\right) \mid c \in C(G_n)\right\}$, *где* c *– цикл из множества* $C(G_n)$ *всех простых циклов в графе* G_n, $l(c)$ *– его длина,* $\nu(c)$ и $\mu(c)$ *– веса входной и выходной разметок этого цикла.*

Здесь для множества *D* точек плоскости через *ConvD* обозначена его выпуклая оболочка. Под расстоянием ρ между двумя точками на плоскости будем понимать максимум модулей разностей координат этих точек.

<u>Теорема 2.</u> Справедливо неравенство

$$\rho\left(\left(p^{(N)}, \pi^{(N)}\right), R_f\right) \le \frac{n}{N+n}. \qquad (2)$$

Мы будем пользоваться двояким представлением цикла: как циклической последовательности различных инцидентных вершин графа, так и циклической двоичной последовательности, в которой все участки длины n различны. Например, циклу в G_n с множеством вершин $\{(0101...),(1010...)\}$ соответствует двоичная (рассматриваемая циклически) последовательность (01).

Для $c \in C(G_n)$ через $z_f(c)$ обозначим вектор $\left[\frac{\|c\|}{l(c)}, \frac{\|f/c\|}{l(c)}\right]$, где

$\|c\| = \sum_{(x_1,x_2,...,x_n)\in c} x_1$ – вес цикла c (число единиц в его двоичной записи),

$\|f/c\| = \sum_{(x_1,x_2,...,x_n)\in c} f(x_1,x_2,...,x_n)$ – вес функции f на вершинах цикла c (обе последние суммы берутся по всем вершинам цикла c),

$l(c)$ – длина цикла c.

В частности, вес и длина цикла $c=(01)$ равны соответственно 1 и 2, $z_f(c) = \left(\frac{1}{2}, \frac{f(0101...)+f(1010...)}{2}\right)$.

Переформулируем теорему 1, чтобы подчеркнуть роль функции f:

Теорема 1.

$$R_f = Conv\left\{\left(\frac{\|c\|}{l(c)}, \frac{\|f(c)\|}{l(c)}\right) \mid c \in C(G_n)\right\}, \qquad (3)$$

где $\|c\|$ – вес цикла c, то есть число единиц в его двоичной записи, $\|f(c)\|$ – вес функции на цикле c, то есть сумма значений функции на вершинах цикла c.

Вычислительная трудоемкость алгоритма построения многоугольника конкретной булевой функции, основанного на непосредственном использовании выражения (3), связана с тем, что количество $|C(G_n)|$ различных циклов графа G_n быстро возрастает с ростом n. В работе Bryant, Christensen[5] приведены следующие

результаты: $|C(G_1)|=3,\ |C(G_2)|=6,\ |C(G_3)|=19,\ |C(G_4)|=179$. Очевидно, величина $|C(G_n)|$ имеет дважды экспоненциальный характер роста. Нижней границей может служить, например, число $2^{2^{n-1}-n}$ полных циклов[6], а в качестве верхней границы можно использовать число 2^{2^n} всех подмножеств множества вершин графа. Поэтому непосредственное использование (3) возможно только при нескольких начальных значениях *n*. Некоторые алгоритмические аспекты построения циклов графов де Брейна освещены в работе Etzion[7].

Простейшие свойства многоугольников булевых функций

Утверждение 1. Пусть $n \geq 2$ *и одна из крайних переменных* (x_1 *либо* x_n) *является несущественной для функции* $f(x_1, x_2, ..., x_n)$, *то есть* $f(x_1, x_2, ..., x_n) = g(x_1, x_2, ..., x_{n-1})$ *либо* $f(x_1, x_2, ..., x_n) = g(x_2, x_3, ..., x_n)$ *для некоторой функции* $g \in F_{n-1}$. *Тогда* $R_f = R_g$. ■

Сформулированное утверждение, легко следующее из определения многоугольника булевой функции (1), теряет свою справедливость в случае, когда несущественной переменной является «внутренняя» переменная x_i, $1 < i < n$. Так, например, из изложенных ниже результатов следует, что многоугольники функций $x_1 \oplus x_2 \oplus x_3$ и $x_1 \oplus x_2 \oplus x_4$ различны (рис. 7, 9).

Многоугольники функций будем изображать в квадрате $0 \leq z_1, z_2 \leq 1$, ось $(0\ z_1)$ – горизонтальна, ось $(0\ z_2)$ – вертикальна.

Утверждение 2. Многоугольники функций $f(x_1, x_2, ..., x_n)$ *и* $f(\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)$ *симметричны относительно вертикальной оси* $z_2 = \frac{1}{2}$. *Многоугольники функций* $f(x_1, x_2, ..., x_n)$ *и* $\overline{f(x_1, x_2, ..., x_n)}$ *симметричны относительно горизонтальной оси* $z_1 = \frac{1}{2}$. *Многоугольник самодвойственной функции центральносимметричен относительно точки* $\left(\frac{1}{2}, \frac{1}{2}\right)$.

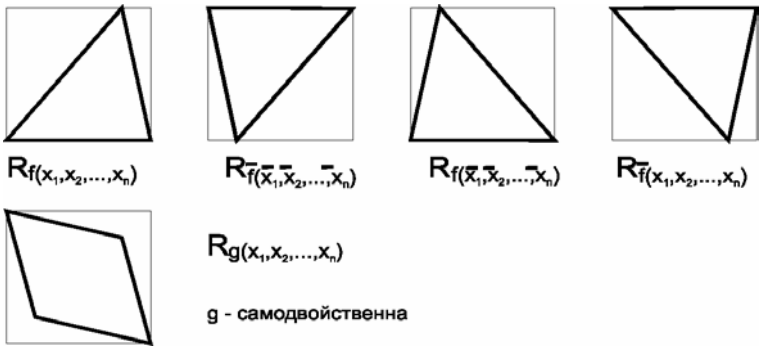


Рис. 1

Доказательство. Докажем, например, первый пункт утверждения. Согласно определению осевой симметрии, точка плоскости $A=(x_A, y_A)$ симметрична точке $B=(x_B, y_B)$ относительно оси $y=\frac{1}{2}$ тогда и только тогда, когда выполняются соотношения $y_A = y_B$ и $x_A = 1 - x_B$. Пусть $g(x_1, x_2, ..., x_n) = f(\bar{x}_1, \bar{x}_2, ..., \bar{x}_n)$. Рассмотрим произвольный цикл $c = (a_1, a_2, ..., a_l) \in C(G_n)$, $a_i = 0,1$, $i = 1,2,...,l$, $l = 1,2,...$ и двойственный к нему $\bar{c} = (\bar{a}_1, \bar{a}_2, ..., \bar{a}_l)$. Имеем:

$$z_g(c) = \left(z_g^{(1)}(c), z_g^{(2)}(c)\right) = \left(\frac{\sum_{i=1}^{l} a_i}{l}, \frac{\left\|g/c\right\|}{l}\right) = \left(\frac{l - \sum_{i=1}^{l} \bar{a}_i}{l}, \frac{\left\|f/\bar{c}\right\|}{l}\right) =$$

$$= \left(1 - z_f^{(1)}(\bar{c}), z_f^{(2)}(\bar{c})\right).$$

Рассмотрим выпуклую оболочку от правой и левой частей последнего равенства:

$$R_g = Conv\left\{\left(z_g(c)\right), c \in C(G_n)\right\} = Conv\left\{\left(1 - z_f^{(1)}(\bar{c}), z_f^{(2)}(\bar{c})\right), c \in C(G_n)\right\}.$$

Учитывая, что множество циклов, двойственных к $C(G_n)$, совпадает с $C(G_n)$, правая часть равна $Conv\left\{\left(1 - z_f^{(1)}(c), z_f^{(2)}(c)\right), c \in C(G_n)\right\}$, то есть является многоугольником, симметричным R_f, что и доказывает первый пункт утверждения. ■

Заметим, что приведенные утверждения в обратную сторону не верны. Например, в F_3 имеются несамодвойственные функции, многоугольники которых симметричны относительно центра квадрата.

Следующее утверждение иллюстрирует связь многоугольника булевой функции с вероятностной функцией[8] автомата A_f. Нетрудно видеть, что если A_f перерабатывает бернуллиевскую последовательность двоичных случайных величин с параметром p, то вероятностная функция $\Phi_f(p)$ этого автомата, значение которой можно интерпретировать как предельное значение вероятности единицы в выходной последовательности, имеет вид

$$\Phi_f(p) = \sum_{i=0}^{n} \left\| f(x_1, x_2, ..., x_n) \Big/ \left\| (x_1, x_2, ..., x_n) \right\| = i \right\| p^i (1-p)^{n-i},$$

где $\left\| f(x_1, x_2, ..., x_n) \Big/ \left\| (x_1, x_2, ..., x_n) \right\| = i \right\|$ – вес функции f на векторах, содержащих ровно i единиц, $i = 0,1,...,n$.

Утверждение 3. Все точки графика вероятностной функции $z_2 = \Phi_f(z_1)$, $z_1 \in [0,1]$ принадлежат многоугольнику R_f.

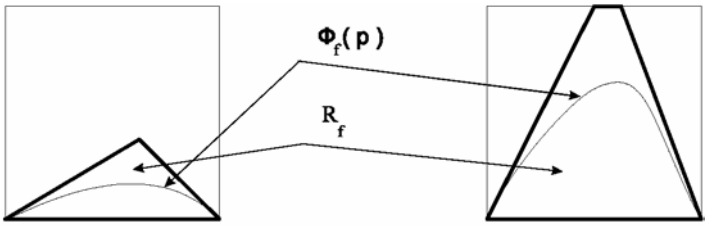


Рис. 2

Доказательство. Обозначим:

$$\tau_i = \binom{n}{i} z_1^i (1-z_1)^{n-i},\ S_i = \left\| f(x_1, x_2, ..., x_n) \Big/ \left\| (x_1, x_2, ..., x_n) \right\| = i \right\|,\ i = 0,1,...,n.$$

Тогда $z_1 = \sum_{i=0}^{n} \frac{i}{n}\tau_i$, $\Phi_f(z_1) = \sum_{i=0}^{n} S_j z_1^i (1-z_1)^{n-i} = \sum_{i=0}^{n} \frac{S_i}{\binom{n}{i}}\tau_i$, следовательно,

$\left(z_1, \Phi_f(z_1)\right) \in Conv\left\{\left(\frac{i}{n}, \frac{S_i}{\binom{n}{i}}\right), i = 0,1,...,n\right\}$. Рассматривая совокупность n-мерных векторов веса i, $i = 0,1,...,n$ как объединение непересекающихся циклов из $C(G_n)$, получаем, что

$$Conv\left\{\left(\frac{i}{n}, \frac{S_i}{\binom{n}{i}}\right), i = 0,1,...,n\right\} \subset Conv\left\{\left(\frac{\|c\|}{l(c)}, \frac{\|f(c)\|}{l(c)}\right) \mid c \in C(G_n)\right\} = R_f. \blacksquare$$

Заметим, что уже в F_3 можно указать как примеры функций с равными многоугольниками, но различными вероятностными функциями, так и, наоборот, примеры функций с совпадающими вероятностными функциями, но с различными многоугольниками.

Утверждение 4. Если $f \in F_n$ – равновероятная функция, т. е. $\|f\| = 2^{n-1}$, $n = 1,2,...$, то

1) $(1/2,1/2) \in R_f$;

2) *Если* $f(0,0,...,0) = f(1,1,...,1)$, *то* $(1/2,1/2)$ – *внутренняя точка* R_f;

3) *Если* $f(0,0,...,0) \neq f(1,1,...,1)$, *то* $(1/2,1/2)$ – *либо внутренняя точка* R_f, *либо она лежит на диагонали квадрата, являющейся стороной многоугольника* R_f.

Доказательство. Первый пункт следует из предыдущего утверждения, поскольку для равновероятной булевой функции f очевидно $\Phi_f(1/2) = 1/2$. Для доказательства п. 2 предположим, что $f(0,0,...,0) = f(1,1,...,1) = 0$. Рассмотрим цикл,

проходящий через все вершины графа G_n, за исключением нулевой и единичной. В многоугольнике R_f этому циклу соответствует точка $\left(1/2,\, 2^{n-1}/\left(2^n-2\right)\right)$. Осталось заметить, что точка $(1/2,1/2)$ является внутренней для треугольника $Conv\left((0,0),(1,0),\left(1/2,\, 2^{n-1}/\left(2^n-2\right)\right)\right)\subseteq R_f$. Третий пункт утверждения доказывается аналогично. ■

Обращение доказанного утверждения неверно. В качестве контрпримера можно указать функцию веса 2, принимающую единичные значения на множестве векторов $\{(0101...),(1010...)\}$. Нетрудно убедиться в том, что многоугольник такой функции является треугольником $Conv\left((0,0),(1,0),(1/2,1)\right)$. Однако при $n>2$ указанная функция неравновероятна.

Оказывается, существуют функции сравнительно большого веса, многоугольники которых целиком лежат ниже верхней стороны квадрата. Майкелтвейтом[9] доказана следующая гипотеза Голомба: минимальное число вершин, которое нужно удалить из графа G_n, чтобы получился граф без циклов, равно

$$\frac{1}{n}\sum_{d/n}\phi(\frac{n}{d})2^d,$$

где через $\phi(n)$ обозначена функция Эйлера. Воспользовавшись представлением (2) для функции, равной нулю на удаляемом множестве и единице на остальных вершинах графа G_n, получаем:

Утверждение 5. Пусть $f\in F_n$, $n=1,2,\ldots$.

Если $\|f\|>2^n-\frac{1}{n}\sum_{d/n}\phi(\frac{n}{d})2^d$, *то в многоугольнике* R_f *имеется точка с ординатой* 1. *Среди функций веса* k, $0\le k\le 2^n-\frac{1}{n}\sum_{d/n}\phi(\frac{n}{d})2^d$ *существуют такие, многоугольники которых целиком лежат ниже прямой* $z_2=1$. ■

Следующий результат показывает, что многоугольники функций веса 1 являются треугольниками. Для *n*-мерного

двоичного вектора $\alpha=(a_1,a_2,...,a_n)$, $n=1,2,...$ через $t=t(\alpha)$ обозначим наименьшее целое число с тем свойством, что

$$\alpha=(a_1,a_2,...,a_n)=(a_1,a_2,...,a_t,...,a_1,a_2,...,a_t,a_1,a_2,...,a_r),\ 0\le r<t.$$

Обозначим $\mu(\alpha)=\|(a_1,a_2,...,a_t)\|$.

Утверждение 6. *Пусть* $(0,0,...,0)\ne\alpha\ne(1,1,...,1)$,

$$f(x_1,x_2,...,x_n)=\begin{cases}1, \textit{если}(x_1,x_2,...,x_n)=\alpha\\ 0, \textit{если}(x_1,x_2,...,x_n)\ne\alpha\end{cases}.$$

Тогда $R_f=Conv\{(0,0),(1,0),(\mu(\alpha)/t(\alpha),1/t(\alpha))\}$. ■

Доказательство проводится тем же методом, что и приводимое ниже доказательство утверждения 12, и поэтому здесь не приводится.

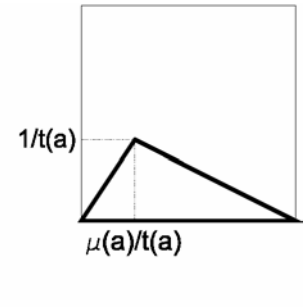


Рис. 3

Заметим, что попытка классификации многоугольников булевых функций по весам этих функций не приводит к успеху: так, при $n=4$ многоугольники функций веса 2 могут иметь три, четыре или пять вершин.

Утверждение 7. *Если* $f\in F_n$, *то*

$$R_f\subseteq Conv\{(0,f(0,0,...,0)),(1/n,\overline{f(0,0,...,0)}),\ (1-1/n,\overline{f(1,1,...,1)}),(1,f(1,1,...,1))\} \quad (4)$$

Доказательство. Ограничимся случаем $f(0,0,...,0)=0$, $f(1,1,...,1)=1$. Нам надо доказать, что произвольная точка $(\xi,\eta)\in R_f$ принадлежит трапеции $\{(0,0),(1/n,1),(1-1/n,0),(1,1)\}$.

Предположим противное. Возможны два варианта: $(\xi,\eta)\in Conv\{(0,0),\ (0,1),(1/n,1)\}$ или $(\xi,\eta)\in Conv\{(1-1/n,0),$ $(\xi,\eta)\in Conv\{(1-1/n,0),\ (1,0),(1,1)\}$. Рассмотрим первый вариант. Должны выполняться неравенства $0<\xi<1/n$, $\eta>n\xi$. В этом случае в множестве $C(G_n)$ найдется такой цикл $c=(\alpha_1,\alpha_2,...,\alpha_l)$, $\alpha_i\in V_n, i=1,2,...,l,$ для которого $z_2(c)>nz_1(c)$. Из последнего неравенства следует:

$$\sum_{i=1}^{l} f(\alpha_i)>n\frac{\sum\|\alpha_i\|}{n}=\sum_{i=1}^{l}\|\alpha_i\|.$$

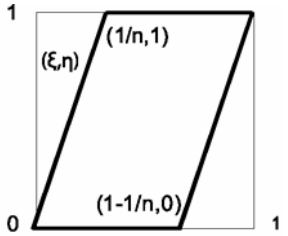


Рис. 4

Если цикл c не проходит через нулевую вершину, то $\sum_{i=1}^{l}\|\alpha_i\|\geq l,$ и получаем противоречие. Если же цикл c проходит через нулевую вершину, то $\sum_{i=1}^{l}\|\alpha_i\|\geq l-1,$ и вновь получаем противоречие, поскольку $f(0,0,...,0)=0$.

Поэтому многоугольник R_f не содержит точек левее границы $z_2=nz_1$. Аналогичными рассуждениями можно показать, что вариант $(\xi,\eta)\in Conv\{(1-1/n,0),(1,0),(1,1)\}$ также невозможен. Оставшиеся случаи значений $f(0,0,...,0)$ и $f(1,1,...,1)$ рассматриваются аналогично. ■

Несложно описать класс функций, многоугольники которых «максимальны». Пусть $R_n^{ab}=Conv\{(0,a),(1/n,\bar{a}),(1-1/n,b),(1,b)\}$, $a,b=0,1$.

Утверждение 8. Для $f \in F_n$ *равенство* $R_f = R_n^{ab}$ *имеет место тогда и только тогда, когда:*

1) $f(0,0,...,0) = a$, $f(1,1,...,1) = b$;

2) *значение* $f(x_1, x_2, ..., x_n)$ *на каждом из векторов веса* 1 *равно* $\bar{a}$, *на каждом из векторов веса* $n-1$ *равно* $\bar{b}$.

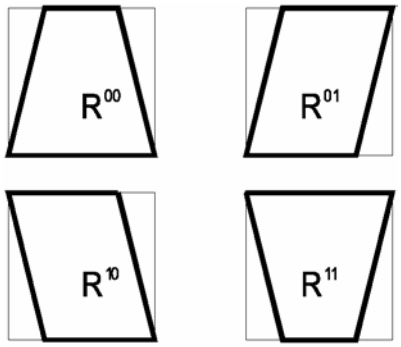


Рис. 5

Доказательство. Достаточность. Пусть f обладает свойствами 1 и 2. Для доказательства того, что $R_f = R_n^{ab}$, достаточно рассмотреть цикл $c = (00...01)$ длины n и двойственный к нему, и применить (4). Для доказательства необходимости условий 1 и 2 рассмотрим случай $a = b = 0$. Пусть $R_f = R_n^{00}$. Очевидно, $f(0,0,...,0) = f(1,1,...,1) = 0$, и f удовлетворяет условию 1. Из того, что точка $(1,1/n)$ является вершиной R_f, следует, что в G_n существует такой цикл $c = (\alpha_1, \alpha_2, ..., \alpha_l)$, $\alpha_i \in V_n, i = 1,2,...,l$, для которого $\sum_{i=1}^{l} f(\alpha_i) = \sum_{i=1}^{l} \|\alpha_i\|$, и, кроме того, на этом цикле функция $f(x_1, x_2, ..., x_n)$ тождественно равна 1. Отсюда $\sum_{i=1}^{l} \|\alpha_i\| = l$, и поскольку $\alpha_i \neq (0,0,...,0)$, то $\|\alpha_i\| = 1$, $i = 1,2,...,l$, что означает, что $l = n$, $c = (00...01)$. Остальные варианты значений a и b рассматриваются аналогично. ■

Приведенный результат относится к тем редким случаям, когда удается описать все функции из F_n с заданным многоугольником. Другой нетривиальный случай соответствует функциям, многоугольники которых «минимальны»[10] (см. также рис. 17). Отметим, что в обоих случаях количество функций, соответствующих одному многоугольнику, растет как двойная экспонента от n.

Утверждение 9. Пусть $f \in F_n$, $n = 3,4,\ldots$, $v(f)$ *обозначает число вершин многоугольника* R_f.

Если $f(0,0,\ldots,0) = f(1,1,\ldots,1)$, *то* $v(f) \le 1 + \sum_{k=1}^{2^n} \phi(k) < \frac{4^n}{2}$,

если $f(0,0,\ldots,0) \ne f(1,1,\ldots,1)$, *то* $v(f) \le 2\sum_{k=1}^{2^n} \phi(k) < 4^n$.

Доказательство. При $f(0,0,\ldots,0) = f(1,1,\ldots,1)$ абсциссы всех вершин многоугольника должны быть различными в силу его выпуклости. Имеем:

$$\left|\left\{\frac{\|c\|}{l(c)}, c \in C(G_n)\right\}\right| \le \left|\left\{\text{несократимые дроби вида } \frac{p}{q}, 1 \le p \le q \le 2^n\right\}\right| + 1,$$

откуда следует первая оценка.

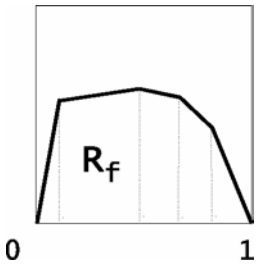


Рис. 6

При $f(0,0,\ldots,0) \ne f(1,1,\ldots,1)$ возможно не более двух вершин с различными абсциссами. ■

Многоугольники некоторых линейных функций и функций, инвариантных относительно циклического сдвига аргументов

Утверждение 10. Справедливо равенство:

$$R_{x_1 \oplus x_2 \oplus \ldots \oplus x_n} = \begin{cases} R_n^{00}, & n - \textit{четно}; \\ R_n^{01}, & n - \textit{нечетно}; \end{cases}$$

Доказательство: по утверждению 8. ■

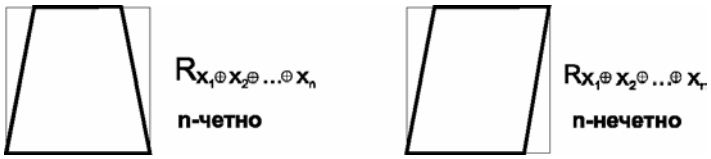


Рис. 7

Утверждение 11. При $n \geq 2$ *справедливо равенство:*

$$R_{x_1 \oplus x_n} = Conv\{(0,0),(1,0),(1/2,1)\}.$$

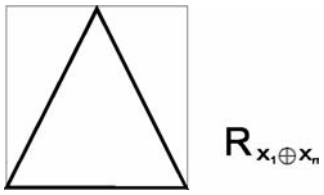


Рис. 8

Доказательство. Поскольку $f(0,0,\ldots,0) = f(1,1,\ldots,1) = 0$ и, кроме того, на цикле $c = (00\ldots011\ldots1)$ длины $2n-2$ и веса $n-1$ функция f тождественно равна единице, то в силу (3) заключаем, что $R_{x_1 \oplus x_n} \supseteq Conv\{(0,0),(1,0),(1/2,1)\}$.

Для доказательства обратного включения покажем, что для произвольного цикла $c = (\alpha_1, \alpha_2, \ldots, \alpha_l)$ из $C(G_n)$ справедливы неравенства:

$$\|f/c\| \leq 2\|c\|, \ \|f/c\| \leq 2l - 2\|c\|.$$

В самом деле,

$$\|f/c\| = \sum_{i=1}^{l}(a_i \oplus a_{i+n-1}) = \sum_{i=1}^{l}(a_i + a_{i+n-1} - 2a_i a_{i+n-1}) = 2\|c\| - 2\sum_{i=1}^{l} a_i a_{i+n-1} \le 2\|c\|.$$

Второе неравенство получается из первого по двойственности, поскольку $x_1 \oplus x_n = \overline{x_1} \oplus \overline{x_n}$. Неравенства доказывают обратное включение и утверждение. ■

Следующее утверждение показывает, что вид многоугольника линейной функции зависит не только от числа ее существенных переменных.

Утверждение 12. Пусть многочлен $a(x) = \sum_{i=0}^{n-1} a_i x^i \in GF(2)[x]$ *является примитивным степени* $n-1$. *При* $n \ge 4$ *число вершин многоугольника* R_f, *соответствующего функции* $f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} a_{i-1} x_i$, *не менее шести. Если* $n = 4$, *то* R_f *– шестиугольник* $Conv\{(0,0), (1/4,3/4), (3/7,1), (4/7,0), (3/4,1/4), (1,1)\}$.

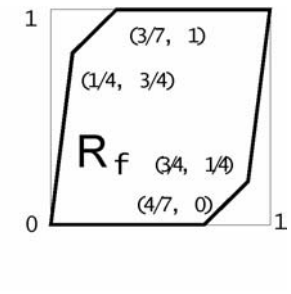


Рис. 9

Рис. 9

Доказательство. Пусть k – число ненулевых коэффициентов многочлена $a(x)$. Очевидно, k нечетно, $k \ge 3$. В силу примитивности $a(x)$ среди $C(G_n)$ только два цикла образованы линейным рекуррентным соотношением $b_{i+n-1} = \sum_{j=0}^{n-2} \oplus a_j b_{i+j}$ – это (0) и цикл c_a, проходящий через все остальные вершины графа.

Заметим, что функция $f(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} a_{i-1} x_i$ равна нулю на тех и только тех вершинах графа G_n, через которые проходят циклы (0) и цикл c_a. Циклу (0) соответствует точка $(0,0)$ множества R_f, а циклу c_a, как нетрудно видеть, точка $\left(2^{n-2}/\left(2^{n-1}-1\right), 0\right)$. Так как других циклов с данным свойством нет, то и правее точки $\left(2^{n-2}/\left(2^{n-1}-1\right), 0\right)$ на прямой $z_2 = 0$ нет точек множества R_f. В силу нечетности k функция f самодвойственна, поэтому многоугольник R_f симметричен относительно точки $(1/2, 1/2)$. Это означает, что на прямой $z_2 = 1$ лежит сторона R_f, которая в точности является отрезком $\left[\left(2^{n-2}-1/\left(2^{n-1}-1\right), 1\right), (1,1)\right]$. Рассмотрим цикл $c = (00...01)$ длины n и веса 1 в $C(G_n)$. Очевидно, $\|f/c\| = k$, и циклу c соответствует точка $(1/n, k/n) \in R_f$.

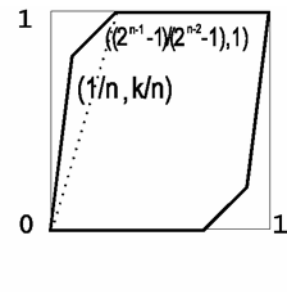


Рис. 10

Неравенство $\frac{k/n}{1/n} = k \geq 3 > \frac{2^{n-1}-1}{2^{n-2}-1}$, справедливое при $n \geq 4$, показывает, что точка $(1/n, k/n) \in R_f$ находится над отрезком, соединяющим точки $(0,0)$ и $\left(2^{n-2}-1/\left(2^{n-1}-1\right), 1\right)$. Однако, как доказано выше, левее точки $\left(2^{n-2}-1/\left(2^{n-1}-1\right), 1\right)$ на прямой $z_2 = 1$ точек R_f нет. Отсюда следует, что помимо точек $(0,0)$ и

$\left(2^{n-2}-1/\left(2^{n-1}-1\right),1\right)$, являющихся вершинами R_f, треугольник $Conv\left\{(0,0),(0,1),\left(2^{n-2}-1/\left(2^{n-1}-1\right),1\right)\right\}$ содержит еще по крайней мере одну вершину R_f, ордината которой отлична от нуля и единицы. В силу симметричности R_f относительно точки $(1/2,1/2)$ число вершин многоугольника не менее шести. При $n=4$ очевидно, $k=3$. Существуют лишь два примитивных многочлена третьей степени: $a_1(x)=1\oplus x\oplus x^3$, $a_2(x)=1\oplus x^2\oplus x^3$. Им соответствуют линейные функции $f_1=x_1\oplus x_2\oplus x_4$, $f_2=x_1\oplus x_3\oplus x_4$. Нам необходимо показать, что точка $(1/4,3/4)$ является вершиной как R_{f_1}, так и R_{f_2}. Этот факт следует из того, что треугольник $Conv\left\{(1/4,3/4),(1/4,1),(3/7,1)\right\}$ не содержит ни одной точки вида $\left(\frac{\|c\|}{l(c)},\frac{j}{l(c)}\right)$, $j=0,...,l(c)-1$ для $c\in C(G_4)$, в чем легко убедиться непосредственно. Поэтому точка $(1/4,3/4)$ – единственная вершина R_{f_1} и R_{f_2}, лежащая выше отрезка $\left[(0,0),(3/7,1)\right]$. ■

<u>Утверждение 13</u>. Пусть $f(x_1,x_2,...,x_n)$ инвариантна относительно циклического сдвига аргументов, т. е. $f(x_1,x_2,...,x_n)=$ $=f(x_2,x_3,...,x_n,x_1)$. Тогда многоугольник R_f является четырехугольником (в вырожденных случаях – треугольником или отрезком):

1) *если* $f(0,0,...,0)=f(1,1,...,1)=0$, *то* $R_f=Conv\left\{(0,0),(1,0),\right.$ $\left.(t_1/n,1),(t_2/n,1)\right\}$,

где t_1 и t_2 – соответственно наименьший и наибольший вес вектора $(x_1,x_2,...,x_n)$, такого, что $f(x_1,x_2,...,x_n)=1$;

2) *если* $f(0,0,...,0)=0$, $f(1,1,...,1)=1$, *то* $R_f=Conv\left\{(0,0),\right.$ $(1,1),(k_1/n,1),\quad\left.(k_2/n,0)\right\}$, *где k_1 – наименьший вес вектора*

$(x_1, x_2, ..., x_n)$, *такого, что* $f(x_1, x_2, ..., x_n) = 1$; k_2 – *наибольший вес вектора* $(x_1, x_2, ..., x_n)$, *такого, что* $f(x_1, x_2, ..., x_n) = 0$.

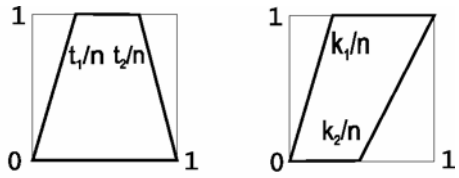


Рис. 11

Доказательство. Ограничимся случаем $f(0,0,...,0) = 0$, $f(1,1,...,1) = 1$. Заметим, что по условию f постоянна на циклах графа G_n, образованных простым сдвиговым регистром. Вектора, через которые проходят эти циклы, имеют один и тот же вес. Поэтому, если $k_1 = \|x_1, x_2, ..., x_n\|$ – вес вектора $(x_1, x_2, ..., x_n) \in V_n$ и $f(x_1, x_2, ..., x_n) = 1$, то на цикле, порожденном циклическими сдвигами вектора $(x_1, x_2, ..., x_n)$, функция f тождественно равна единице. Если r – вес этого цикла, а l – длина, то, как легко видеть, $r/l = k_1/n$. Поэтому $(k_1/n, 1) \in R_f$. Аналогично и $(k_2/n, 0) \in R_f$. Отсюда $R_f \supseteq Conv\{(0,0),(1,1),(k_1/n,1),(k_2/n,0)\}$.

Для того чтобы доказать обратное включение, покажем, что все точки множества R_f лежат не левее прямой (a), уравнение которой есть $z_2 = \frac{n}{k_1} z_1$.

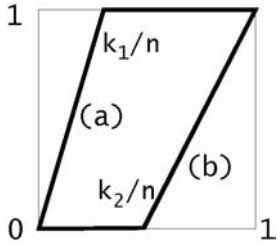


Рис. 12

Достаточно показать, что для любого цикла $c \in C(G_n)$, множество вершин которого есть $V = \{\alpha_1, \alpha_2, ..., \alpha_l\}$, выполняется неравенство:

$$k_1 \sum_{\alpha \in V} f(\alpha) \le \sum_{\alpha \in V} \|\alpha\| \qquad (5)$$

Пусть A_i – множество всех n-мерных двоичных векторов веса i, $i = 0,1,...,n$. Тогда, с одной стороны,

$$k_1 \sum_{\alpha \in V} f(\alpha) \le \sum_{j=k_1}^{n} |A_j \cap V|;$$

с другой стороны,

$$\sum_{\alpha \in V} \|\alpha\| = \sum_{j=0}^{n} j |A_j \cap V| = \sum_{j=0}^{k_1-1} j |A_j \cap V| + \sum_{j=k_1}^{n} j |A_j \cap V| \ge k_1 \sum_{j=k_1}^{n} |A_j \cap V|,$$

доказывая (5).

Покажем, что все точки множества R_f лежат не правее прямой (b), уравнение которой есть $(n - k_2 + 1) z_2 = n z_1 - (k_2 - 1)$.

Для этого достаточно показать, что для произвольного цикла $c \in C(G_n)$ с множеством вершин $V = \{\alpha_1, \alpha_2, ..., \alpha_l\}$ выполняется неравенство

$$(n - k_2 + 1) \sum_{\alpha \in V} f(\alpha) \ge \sum_{\alpha \in V} \|\alpha\| - l(k_2 - 1) \qquad (6)$$

Имеем, с одной стороны, $(n - k_2 + 1) \sum_{\alpha \in V} f(\alpha) \ge (n - k_2 + 1) \sum_{j=k_2}^{n} j |A_j \cap V|$;

с другой стороны, поскольку $l = \sum_{j=0}^{n} |A_j \cap V|$, то

$$\sum_{\alpha \in V} \|\alpha\| - l(k_2 - 1) = \sum_{j=0}^{n} j |A_j \cap V| - \sum_{j=0}^{n} (k_2 - 1) |A_j \cap V| = \sum_{j=0}^{n} (j - k_2 + 1) |A_j \cap V| =$$

$$= \sum_{j=k_2}^{n} (j - k_2 + 1) |A_j \cap V| + \sum_{j=0}^{k_2-1} (j - k_2 + 1) |A_j \cap V| \le (n - k_2 + 1) \sum_{j=k_2}^{n} j |A_j \cap V|,$$

доказывая (6). Объединяя (5) и (6), получаем, что $R_f \subseteq Conv\{(0,0),(1,1),(k_1/n,1),(k_2/n,0)\}$. ■

Следствие 1. *Если* $f_k(x_1, x_2, ..., x_n)$ – *пороговая функция вида*

$$f_k(x_1, x_2, ..., x_n) = \begin{cases} 1, \text{ если } \left\| (x_1, x_2, ...x_n) \right\| \geq k \\ 0 \quad \text{в противном случае} \end{cases}, \text{ то}$$

$$R_f = Conv\{(0,0),(1,1),(k-1/n,0),(k/n,1)\} \qquad (7)$$

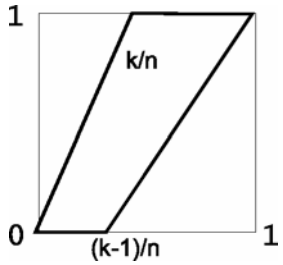


Рис. 13

Следствие 2.

$$R_{x_1 x_2 ... x_n} = Conv\{(0,0),(1,1),(n-1/n,0)\},$$

Рис. 1

$$R_{x_1 \vee x_2 \vee ... \vee x_n} = Conv\{(0,0),(1,1),(1/n,1)\}. \blacksquare$$

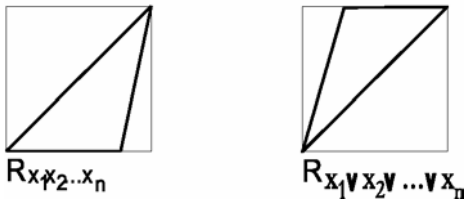


Рис. 14

Многоугольники функций малого числа переменных. Примеры

Все четыре функции **одного** переменного имеют различные многоугольники (константа «0» – $[(0,0),(1,0)]$, константа «1» – $[(0,1),(1,1)]$, функция «x_1» – $[(0,0),(1,1)]$, функция «$\overline{x_1}$» – $[(0,1),(1,0)]$). Вычисления показывают, что функциям **двух**

переменных соответствуют 12 различных многоугольников. Имеется 68 различных многоугольников функций **трех** переменных. Имеется 1520 различных многоугольников функций **четырех** переменных, среди которых 4 отрезка, 106 треугольников, 576 четырехугольников, 662 пятиугольника, 164 шестиугольника и 8 семиугольников. Максимальное число функций соответствует треугольнику *Conv* {(0,0), (1/2,1), (1,0)} и симметричному ему (по 1989 каждому). Отметим неожиданно большое (22) число функций, многоугольник которых есть отрезок $\left[(0,0),(1,1)\right]$ (рис. 17).

Приводимые ниже примеры иллюстрируют соотношение (2) для различных значений *N*. Компьютерный эксперимент состоял в генерации псевдослучайной двоичной последовательности длины *L*, которая подавалась на на вход автомата A_f. Затем эта последовательность вместе с выходной последоватльностью «нарезалась» на отрезки длины N с шагом 1, т. е. с перекрытием соседних на $N-1$, так что общее число пар отрезков равно $L-N+1-n$, как показано на рис. 15.

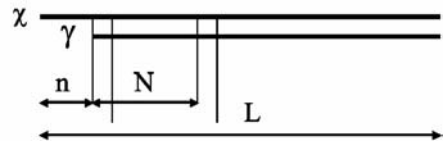


Рис. 15

Для каждого такого отрезка составлялась пара чисел $\left(\frac{\textit{вес входа}}{N},\frac{\textit{вес выхода}}{N}\right)$ и отображалась точкой в квадрате $[0,1]\times[0,1]$.

На рис. 16 представлены изображения многоугольника пороговой функции $f(x_1,x_2,x_3,x_4)=\begin{cases}1, \textit{если } x_1+x_2+x_3+x_4\geq 2\\ 0 \quad \textit{в противном случае}\end{cases}$, который, согласно (7), имеет вид $Conv\{(0,0),(1/4,0),(1/2,1),(1,1)\}$ для $L=10000$, длина N рабочего отрезка изменяется в пределах от 5 до 500.

На рис. 17 для диапазона значений $L = 500 \div 50000$ представлены многоугольники нескольких функций четырех переменных, иллюстрирующие утверждения 10–13.

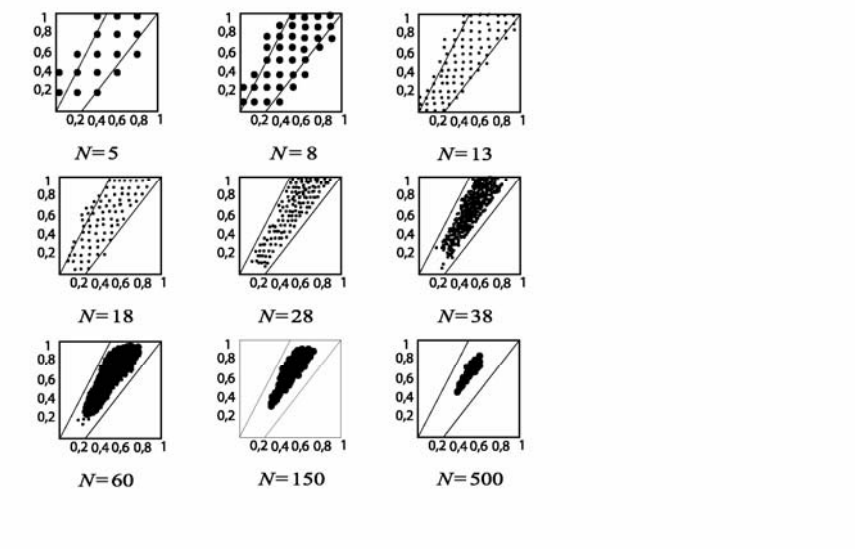


Рис. 16

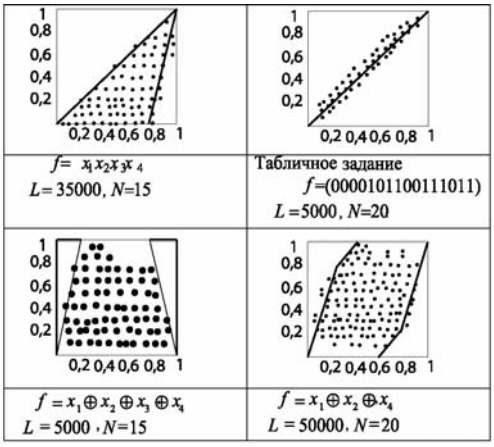


Рис. 17

Примечания

[1] См.: *Никонов В.Г.* Классификация минимальных базисных представлений всех булевых функций от четырех переменных // Обозрение прикладной и промышленной математики. М., 1994. Т. 1. Вып. 3; *Логачев О.А., Сальников А.А., Ященко В.В.* Булевы функции в теории кодирования и криптологии. М.: МЦНМО, 2004.

[2] *Мельников С.Ю.* Многогранники, характеризующие статистические свойства конечных автоматов // Труды по дискретной математике. М.: Изд-во физико-математической литературы, 2003. Т. 7. С. 126–137.

[3] См.: *Мельников С.Ю.* Регистры сдвига с чезаровским входом // Вестник Московского государственного ун-та леса. Лесной вестник. 2005. № 4 (35).

[4] См.: Там же.

[5] *Bryant P.R., Christensen J.* The enumeration of shift register sequences // Journal of Comb. Theory. Ser. A. 1983. Vol. 35. P. 154–172.

[6] См.: *Холл М.* Комбинаторика. М.: Мир, 1970.

[7] *Etzion T.* An algorithm for generating shift – register cycles // Theor. Comp. Sci. 1986. Vol. 44. P. 209–224.

[8] См.: *Кудрявцев В.Б., Алешин С.В., Подколзин А.С.* Введение в теорию автоматов. М.: Наука, 1985.

[9] *Mykkeltveit J.* A proof of Golomb`s conjecture for the de Bruijn graph // Journal of Comb. Theory. Ser. B. 1972. Vol. 13. P. 40–45.

[10] *Мельников С.Ю.* О классе двоичных функций, сохраняющих значковые статистические свойства последовательностей при преобразовании сдвигового типа // Обозрение прикладной и промышленной математики. М., 2007. Т. 14. Вып. 6. С. 1123–1124.

А.А. Липатьев

ПОНИЖЕНИЕ РАЗМЕРНОСТИ И КЛАССИФИКАЦИЯ

В статье рассмотрено понижение размерности методом главных компонент и кластеризация методом К-средних. Для данной комбинации методов доказана принципиальная возможность проведения кластеризации методом К-средних для данных, являющихся результатом понижения размерности исходного набора данных.

Ключевые слова: метод К-средних, метод главных компонент, многомерный анализ.

В данной статье описывается способ изучения корректности классификационных решений при снижении размерности.

При кластеризации данных различного рода часто возникают сложности, связанные с большой размерностью данных. В таких случаях естественным образом встает вопрос: а нельзя ли безболезненно заменить десяток параметров, которыми описываются объекты, например, парой самых важных? Если бы мы каким-то образом сделали подобную замену и провели некоторую обработку данных, могли бы мы вообще применять методы, которые можно применять к исходным данным?

В качестве метода классификации в данной статье рассматривается метод К-средних. В качестве метода понижения размерности рассматривается метод главных компонент.

Для применения метода К-средних (точнее для сходимости в этом методе) должны быть выполнены некоторые предположения относительно распределения данных. В данной статье исследуется возможность применения этого метода к набору данных, представляющих собой результат понижения размерности исходных данных.

Детали методов и причины выбора именно этих методов описаны ниже.

Описание используемых методов

Метод К-средних

Метод К-средних является одним из наиболее распространенных и известных методов кластеризации. Это обусловлено простотой применения и интерпретации его результатов.

Впоследствии будет описано, как данный метод применяется на практике, а сейчас перейдем к математическому описанию метода[1].

Считается, что задана последовательность $z_1, z_2, \ldots$ независимых одинаково распределенных случайных векторов, заданных на одном вероятностном пространстве **(Ω, *R*, *P*)** и принимающих значения в $\left(R^N, \Im\right)$, где $\Im$ – σ-алгебра борелевских множеств на R^N. Также случайные величины z_i, $i \in N$ имеют распределение $P_1(A) = P\{z_1 \in A\}$, $A \in \Im$.

Пусть у нас есть так называемый k-кортеж $x = (x_1, \ldots, x_k)$, где k – некоторое натуральное число. Тогда определим *минимальное разбиение* $S(x) = \{S_1(x), S_2(x), \ldots, S_k(x)\}$, $S_j(x) \subset R^N$, $j = \overline{1,k}$ следующим образом:

$$S_1(x) = T_1(x),\ S_2(x) = T_2(x) S_1^C(x), \ldots,$$
$$S_2(x) = T_k(x) S_1^C(x) S_2^C(x) \ldots S_1^C(x),$$

где $T_i(x) = \left\{\xi : \xi \in R^N,\ \|\xi - x_i\| \le \|\xi - x_j\|,\ \forall\ \ j = 1, \ldots, k\right\}$ и

$A^C = \left\{\xi : \xi \in R^N,\ \xi \notin A\right\}$ – дополнение множества A.

Таким образом, множество $S_i(x)$ содержит точки из R^N, ближайшие к x_i, исключая точки, уже включенные во множество из разбиения с меньшим индексом. В силу этого, если $x_i = x_j$ и $i < j$, то $S_i(x) = \varnothing$.

Последовательность k-кортежей $x^n = \left(x_1^n, x_2^n, \ldots, x_k^n\right)$, $x_i^n \in R^N$, $i = 1, \ldots, k$, $n \in N$ и соответствующих им целочисленных весов $\left(w_1^n, w_2^n, \ldots, w_k^n\right)$ определяется следующим образом:

$$x_i^1 = z_i,\ w_i^1 = 1,\ j = 1, \ldots, k \text{ и для } n = 1, 2, \ldots,$$

если $z_{k+n} \in S_i^n$, то $x_i^{n+1} = \frac{x_i^n w_i^n + z_{n+k}}{w_i^n + 1}$, $w_i^{n+1} = w_i^n + 1$ и $x_j^{n+1} = x_j^n$, $w_j^{n+1} = w_j^n$ для $j \neq i$. Здесь $S^n = \{S_1^n, S_2^n, \ldots, S_k^n\}$ – минимальное разбиение, отвечающее x^n.

Другими словами, процедура К-средних состоит в том, что мы берем k групп, каждая из которых состоит из 1 точки, и затем добавляем каждую новую точку к той группе, среднее значение которой ближе к этой точке. После добавления точки среднее значение группы пересчитывается.

Далее требуются некоторые предположения о свойствах распределения векторов z_i, они следующие:

(i) P_1 абсолютно непрерывна относительно меры Лебега на R^N

(ii) $P_1(U) = 1$ для некоторого выпуклого замкнутого ограниченного множества $U \in \Im$ и $P_1(A) > 0$ для любого открытого множества $A \subset U$.

Теперь определим для k-кортежа $x = (x_1, \ldots, x_k)$ следующие две величины:

$$W(x) = \sum_{i=1}^{k} \int_{S_i} \|z - x_i\|^2 \, dP_1,$$

$$V(x) = \sum_{i=1}^{k} \int_{S_i} \|z - u_i(x)\|^2 \, dP_1,$$

где $S = \{S_1, S_2, \ldots, , S_k\}$ – это минимальное разбиение, отвечающее x, а $u_i(x) = \frac{\int_{S_i} z dP}{P(S_i)}$ или $u_i(x) = x_i$, если, соответственно, $P(S_i) > 0$ или $P(S_i) = 0$.

Если $x_i = u_i(x)$, $i = 1, \ldots, k$, то назовем k-кортеж x *несмещенным.*
Основной результат дает следующая:

Теорема 1. Последовательность случайных величин $W(x^1)$, $W(x^2), \ldots$ сходится п.н. и предел $W_\infty = \lim_{n \to \infty} W(x^n)$ равен п.н. $V(x)$ для некоторого x из класса k-кортежей $x = (x_1, \ldots, x_k)$ таких, что x несмещенный и $x_i \neq x_j$, если $i \neq j$. Причем $x^n \to x$ при $n \to \infty$.

Эта теорема означает, что функция $W(x)$, которую можно назвать суммарной внутригрупповой дисперсией, при применении описанной процедуры стремится к локальному минимуму. Этот факт и позволяет использовать метод К-средних для классификации многомерных данных, так как кажется логичным, что при реальном наличии нескольких обособленных групп суммарная внутригрупповая изменчивость будет мала.

Как же этот метод используется на практике?

На практике у исследователя есть набор объектов, представляемых векторами чисел. Также у него есть число k – заранее определенное число групп, на которые следует разделить всю совокупность объектов (откуда берется это число, нам неважно). Для разделения берутся некоторые k точек и считаются отдельными группами, после этого для каждой из оставшихся точек проводится следующая процедура.

Точка присоединяется к той группе, среднее которой ближе к этой точке. В итоге мы получаем k групп точек, объявляемых нами центрами кластеров.

Метод главных компонент

Метод главных компонент является эмпирическим методом снижения размерности. В соответствии с этим методом мы выбираем, сколько и какие числа могут достаточно точно передать структуру наших данных.

Считается, что в распоряжении исследователя находится таблица (матрица) данных:

$Y = \|y_{il}\|_{n\times m}$, где строки соответствуют объектам (их n), а столбцы – признакам, описывающим объекты (их m).

Далее обозначим средние значения $\overline{y}_l = \frac{1}{n}\sum_{i=1}^{n} y_{il}$, $l = \overline{1,m}$ и центрируем данные: $y'_{il} = y_{il} - \overline{y}_l$, $i = \overline{1,n}$, $l = \overline{1,m}$. После этого мы будем считать данные центрированными: $\overline{y}_l = 0$, $l = \overline{1,m}$.

Обозначим через $\hat{\Sigma} = \|\hat{\sigma}_{kl}\|_{m\times m} = \frac{1}{n}Y^T Y$ выборочную ковариационную матрицу (центрированных признаков). То есть $\hat{\sigma}_{kl}$ – выборочная ковариация k-го и l-го столбцов матрицы Y.

Матрица $\hat{\Sigma}$ является неотрицательно определенной. Следовательно, существует ортогональная матрица C ($C^T C = I$), приводящая $\hat{\Sigma}$ к главным осям: $C^T \hat{\Sigma} C = \Lambda$. Здесь Λ – диагональная матрица с неотрицательными элементами $\lambda_1 \geq \cdots \geq \lambda_m \geq 0$ на главной

диагонали, которые являются корнями уравнения $\det\left(\hat{\Sigma} - \lambda I\right) = 0$, где I – единичная матрица. Числа λ_i, $i = \overline{1,m}$ называются собственными значениями матрицы $\hat{\Sigma}$. Предположим, что все λ_i положительны и различны (для реальных данных это условие выполняется практически всегда).

При этом столбцы $c_1,\ldots,c_m$ матрицы C (главные оси или компоненты) определяются однозначно с точностью до выбора направления оси (одновременного изменения знака всех координат вектора c_l). Они образуют новый ортонормированный базис в R^m, обладающий рядом важных свойств. Нам важны лишь некоторые из них[2].

1) Проекции объектов на первую главную компоненту c_1 имеют наибольшую выборочную дисперсию среди проекций на всевозможные направления d в пространстве R^m.

Как это можно себе представить? Рассмотрим рис. 1.

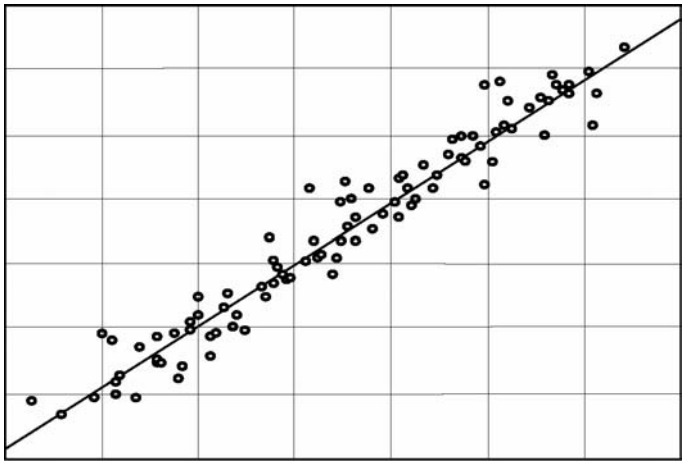


Рис. 1. Пример для размерности 2

На этом рисунке точки группируются вокруг прямой $y = x$. Порой можно считать, что отклонения от этой прямой весьма хаотичны и являются результатом действия неучтенных факторов (шумом). В направлении этой прямой разброс как раз максимален.

2) При $l \geq 2$ вектор c_l является направлением с наибольшей выборочной дисперсией среди направлений, ортогональных векторам $c_1,\ldots,c_{l-1}$.

3) Пусть M_p – это подпространство, натянутое на главные оси $c_1,\ldots,c_p$. При проецировании объектов на произвольное подпространство L_p размерности p в R^m геометрическая структура искажается в наименьшей степени, если этим подпространством является M_p, а именно: сумма квадратов расстояний от объектов до их проекций на L_p минимальна, когда $L_p = M_p$ и равна $n\left(\lambda_{p+1} + \cdots + \lambda_m\right)$. Таким образом, средний квадрат расстояния от объекта до его проекции будет равен $\rho = \frac{1}{n} n\left(\lambda_{p+1} + \cdots + \lambda_m\right) = \lambda_{p+1} + \cdots + \lambda_m$.

Метод главных компонент является весьма распространенным методом понижения размерности. В частности, этот метод применяется при классификации (выявлении однородных групп).

Применение метода главных компонент обусловлено следующими соображениями. Если в рассматриваемой совокупности объектов действительно присутствуют несколько обособленных групп, то выборочная дисперсия в направлении, соединяющем центры групп, наверное, должна быть велика. Таким образом, проецирование на плоскость двух первых компонент (к примеру) может помочь определиться с числом групп.

Данная статья и посвящена вопросу законности применения метода главных компонент к задачам классификации.

Совместное использование методов

Допустим, мы хотим вместо исходных объектов классифицировать их проекции на подпространство, натянутое на несколько первых главных компонент. В этом случае возникает вопрос: будет ли сходимость в методе К-средних?

Этим вопросом мы сейчас и займемся.

Сходимость в методе К-средних

Пусть у нас есть последовательность случайных векторов $z_1, z_2, \ldots$ со значениями в R^m. Эти векторы имеют распределение P_1.

Мы нашли главные компоненты $c_1,\ldots,c_m$ по имеющимся у нас данным. Эти векторы образуют ортонормированный базис в R^m. Теперь мы переходим от исходных векторов к их проекциям на первые l ($l<m$) главных компонент:

$f(z)=\left(\langle z,c_1\rangle_{R^m},\ldots,\langle z,c_l\rangle_{R^m}\right)$, где $\langle\cdot,\cdot\rangle_{R^m}$ – скалярное произведение, а через $f(\cdot)$ обозначен оператор проектирования.

Если мы с помощью этого оператора перейдем от исходной последовательности $z_1,z_2,\ldots$ к последовательности проекций $z_1',z_2',\ldots$, то эти проекции будут иметь другое распределение.

Для исходного распределения P_1 были выполнены следующие требования:

1) Распределение P_1 абсолютно непрерывно относительно меры Лебега.

2) Существует выпуклое замкнутое ограниченное множество $U\subset R^m$ такое, что:

а) $P_1(U)=1$.

б) $P_1(A)>0$ для любого открытого множества $A\subset U$.

Случайные вектора $z_i'=f(z_i)$ будут иметь распределение P_2. Будут ли выполнены два указанных выше условия? Да, будут, и мы это сейчас покажем.

В этой части будем считать, что главные компоненты совпадают с базисом исходного пространства. Это не ограничивает общности, так как мера Лебега инвариантна относительно движений пространства, и при повороте выпуклое замкнутое ограниченное множество таковым и останется. Также при переходе к новому ортонормированному базису сохраняются расстояния между точками.

Абсолютная непрерывность

Итак, нам нужно доказать, что распределение P_2 абсолютно непрерывно относительно меры Лебега на R^l. Возьмем некоторое множество $N\subset R^l$ нулевой меры.

По определению P_2 получаем $P_2(N)=P_1\left(f^{-1}(N)\right)$, где $f^{-1}(N)=\left\{z: z\in R^m,\ f(z)\in N\right\}$ есть полный прообраз множества N.

В силу сделанного нами допущения $f^{-1}(N)$ будет представлять собой «цилиндр», у точек которого вектор из первых l коорди-

нат принадлежит множеству N, а остальные координаты фиксированы.

Покажем, что мера Лебега множества $f^{-1}(N)$ равна 0. Для этого введем множества E_n, $n=1,2,\ldots$ такие, что

$$E_n=\left\{x:\, x=(x_1,\ldots,x_m)\in R^m,\ |x_{l+1}|<n,\ldots,|x_m|<n\right\}.$$

Теперь, если обозначить $N_n=f^{-1}(N)\cap E_n$, очевидно,

$$f^{-1}(N)=\bigcap_{i=1}^{\infty}N_n \ \text{ и } \ N_n<N_{n+1}.$$

В силу непрерывности и определения меры Лебега получаем:
$\lambda\left(f^{-1}(N)\right)=\lim\limits_{n\to\infty}\lambda(N_n)=\lim\limits_{n\to\infty}\lambda(0)=0$, где $\lambda(\cdot)$ – мера Лебега.

Так как мера P_1 абсолютно непрерывна относительно меры Лебега на R^m, то $P_1\left(f^{-1}(N)\right)=0$ и, следовательно, $P_2(N)=0$.

Мы доказали абсолютную непрерывность распределения P_2.

Существование необходимого для сходимости носителя меры P_2

Нам нужно доказать существование выпуклого замкнутого ограниченного множества $U'\subset R^l$ такого, что:

а) $P_2(U')=1$,

б) $P_2(A)>0$ для любого открытого множества $A\subset U'$.

Возьмем в качестве U' следующее множество: $U'=f(U)$.

Оператор проектирования f линеен, то есть $f(\alpha x+\beta y)=$
$=\alpha f(x)+\beta f(y)\quad \forall x,y\in R^m,\ \forall\alpha,\beta\in R.$ Поэтому, если $x',y'\in U'$ и $x,y\in U$ таковы, что $f(x)=x'$ и $f(y)=y'$, то $\forall\lambda\in[0;1]$ имеем
$\lambda x+(1-\lambda)y\in U\Rightarrow\lambda x'+(1-\lambda)y'=\lambda f(x)+(1-\lambda)f(y)=$
$=f\left(\lambda x+(1-\lambda)y\right)\in\dot{U}'$, следовательно множество U' замкнуто.

Покажем, что $\left\|f(x)-f(y)\right\|_{R^l}\le\left\|x-y\right\|_{R^m}\quad\forall x,y\in R^m$. По нашему соглашению, оператор f действует следующим образом:

если $z=(z_1,\ldots,z_m)\in R^m$, то $f(z)=(z_1,\ldots,z_l)\in R^l$, поэтому если $x=(x_1,\ldots,x_m),\ y=(y_1,\ldots,y_m)\in R^m$,

то $\|f(x)-f(y)\| = \sqrt{\sum_{i=1}^{l}(x_i-y_i)^2} \le \sqrt{\sum_{i=1}^{m}(x_i-y_i)^2} = \|x-y\|$.

Из этого можно сделать вывод, что оператор f непрерывен, то есть $\forall z \in R^m \ \ \forall \varepsilon > 0 \ \ \exists \delta > 0$ такое, что $\forall w: \|w-z\|_{R^m} < \delta$ имеет место неравенство $\|f(w)-f(z)\|_{R^l} < \varepsilon$.

Теперь, если число r таково, что $\forall z \in U: \|z\| < r$, то $\forall z' \in U'$ будет выполнено $\|z'\| < r$, следовательно, множество U' ограничено.

Теперь докажем замкнутость множества U'. Пусть последовательность $z_1', z_2', \ldots$ точек из U' сходится к некоторой точке $z' \in R^l$. Тогда существует последовательность точек $z_1, z_2, \ldots$ из U такая, что $f(z_i) = z_i'$. Так как множество U ограничено, то у последовательности $z_1, z_2, \ldots$ существует сходящаяся подпоследовательность $z_{n_1}, z_{n_2}, \ldots$, предел которой мы обозначим через z, причем $z \in U$. Тогда $\lim\limits_{i\to\infty} f(z_{n_i}) = f(z) = \lim\limits_{i\to\infty} z_{n_i}' = \lim\limits_{i\to\infty} z_i' = z'$, поэтому $z' \in U'$, и множество U' замкнуто.

Теперь нужно показать, что для любого открытого множества $A: A \in U'$ будет выполнено $P_2(A) > 0$.

Перед этим представим ряд лемм, доказательство которых вынесено в отдельную часть.

Пусть множество $A \subset U'$ открыто. Через $\operatorname{int} B$ мы будем обозначать внутренность множества B, то есть множество $\{z: z \in B, \ \exists \varepsilon > 0: B_\varepsilon(z) \subset B\}$ – всех внутренних точек множества B, где $B_\varepsilon(z) = \{y: y \in R^l, \|y-z\| < \varepsilon\}$ – открытый шар с центром в точке z и радиусом ε.

Лемма 1. Образом множества $\operatorname{int} U$ при отображении f будет являться открытое множество.

Лемма 2. Множество U содержит хотя бы одну внутреннюю точку.

Лемма 3. Если z' – граничная точка множества U, то существует последовательность $z_1, z_2, \ldots$ внутренних точек множества U, сходящаяся к точке z'.

Лемма 4. Замыкание множества $f(\operatorname{int} U)$ совпадает со множеством U'.

Лемма 5. Множества $\operatorname{int} U$, $f(\operatorname{int} U)$ и $\operatorname{int} U'$ выпуклые.

Доказательство:

Обозначим T_1 – множество всех открытых подмножеств множества $\mathrm{int}U$, тогда пара $(\mathrm{int}U, T_1)$ по определению образует топологическое пространство. Аналогичным образом зададим топологическое пространство $(\mathrm{int}(U'), T_2)$.

По лемме 1 множество $f(\mathrm{int}U)$ открыто, поэтому оно является подмножеством множества $\mathrm{int}U'$, так как $f(\mathrm{int}U)$ состоит только из внутренних точек множества U'. Этот факт позволяет нам рассматривать f как отображение из $(\mathrm{int}U, T_1)$ в $(\mathrm{int}(U'), T_2)$. Отображение f непрерывно, поэтому по свойству непрерывных отображений (в общем случае принимаемому за определение) прообраз любого множества $A \in T_2$ будет принадлежать T_1.

Если мы теперь покажем, что прообраз любого непустого множества $A \in T_2$ не является пустым множеством, то будет доказано, что для любого открытого подмножества C множества U' выполнено $P_2(C) > 0$, то есть U' удовлетворяет всем указанным выше требованиям и в методе К-средних будет иметь место сходимость.

Мы покажем, что $f(\mathrm{int}U) = \mathrm{int}U'$, из этого очевидным образом будет следовать доказываемый факт.

Обозначим через $[A]$ замыкание множества A, то есть пересечение всех замкнутых множеств, содержащих A. Известно, что $[A]$ получается из A добавлением всех предельных точек множества A, то есть таких точек $z \in A$, что существует последовательность $z_1, z_2, \ldots \in A$ такая, что $z_n \to z$ при $n \to \infty$.

По лемме 4 выполнено $\left[f(\mathrm{int}U) \right] = U' = [\mathrm{int}U']$. Допустим, что $f(\mathrm{int}U) \neq \mathrm{int}U'$ и пусть $z \in f(\mathrm{int}U)$ и $z \notin \mathrm{int}U'$.

Раз $z \notin \mathrm{int}U'$ и $z \in f(\mathrm{int}U) \subset U'$, то z лежит на границе множества $\mathrm{int}U'$. Тогда z также лежит на границе множества U'. Так как по лемме 1 $f(\mathrm{int}U)$ является открытым множеством, то z является внутренней точкой множества U' – получили противоречие, поэтому $f(\mathrm{int}U) \subset \mathrm{int}U'$.

Пусть теперь $z \in \operatorname{int} U'$ и $z \notin f(\operatorname{int} U)$. Тогда z лежит на границе множества $f(\operatorname{int} U)$.

Так как множество $f(\operatorname{int} U)$ выпукло, то выполнено[3]:

существует ненулевой вектор $p \in R^c$ такой, что $\langle p, z \rangle_{R^l} \le \langle p, y \rangle_{R^l}$, $\forall y \in f(\operatorname{int} U)$, то есть множество $f(\operatorname{int} U)$ лежит в одном из полупространств, образованных гиперплоскостью $\langle p, z \rangle_{R^l} = \langle p, y \rangle_{R^l}$. Очевидно, что полупространство является замкнутым множеством, следовательно $\left[f(\operatorname{int} U) \right] = U'$ также содержится в этом полупространстве. Точка z лежит на этой гиперплоскости, поэтому по определению является граничной точкой – получили противоречие, поэтому $\operatorname{int} U' \subset f(\operatorname{int} U)$.

Итак, мы показали включение в обе стороны, поэтому $\operatorname{int} U' = f(\operatorname{int} U`)$. Тогда $\forall z' \in \operatorname{int} U'$ существует вектор $z \in \operatorname{int} U$ такой, что $f(z) = z'$. Тем самым мы показали, что прообраз любого непустого множества $A \in T_2$ не является пустым множеством.

Таким образом, для меры P_2 выполнены следующие требования:

1) Распределение P_2 абсолютно непрерывно относительно меры Лебега.

2) Существует выпуклое замкнутое ограниченное множество $U' \subset R^l$ такое, что:

а) $P_2(U') = 1$,

б) $P_2(A) > 0$ для любого открытого множества $A \subset U'$.

Поэтому, если мы применим метод К-средних к последовательности $f(z_1), f(z_2), \ldots$, то будет сходимость, и центры групп в итоге стабилизируются в (описанном выше смысле).

Доказательства

Доказательство леммы 1. Пусть точка $z' \in f(\operatorname{int} U)$, тогда существует точка $z \in \operatorname{int} U$ такая, что $f(z) = z'$. Также для некоторого $\varepsilon > 0$ найдется шар $B_\varepsilon^m(z) = \{ y : y \in R^m, \|y - z\| < \varepsilon \}$ такой, что $B_\varepsilon^m(z) \subset U$.

Проверим, что образом шара $B_\varepsilon^m(z)$ при отображении f будет шар $B_\varepsilon^l(f(z))=\{y: y\in R^l,\ \|y-f(z)\|<\varepsilon\}$. Напомним, что в этой части мы считаем, что отображение проектирования действует следующим образом: если $z=(z_1,\ldots,z_m)\in R^m$, то $f(z)=(z_1,\ldots,z_l)\in R^l$.

Теперь, если $u=(u_1,\ldots,u_m)\in R^m$, $u\in B_\varepsilon^m(z)$, то $(u_1-z_1)^2+\cdots+(u_m-z_m)^2<\varepsilon^2$, тогда $(u_1-z_1)^2+\cdots+(u_l-z_l)^2<\varepsilon^2$. Следовательно, $f(u)=(u_1,\ldots,u_l)\in B_\varepsilon^l(f(z))$, так как $f(z)=(z_1,\ldots,z_l)$.

Если $u'=(u_1,\ldots,u_l)\in R^l$, $u\in B_\varepsilon^l(f(z))$, то $(u_1-z_1)^2+\cdots+(u_l-z_l)^2<\varepsilon^2$ и для $u=(u_1,\ldots,u_l,z_{l+1},\ldots,z_m)$ выполнено соотношение

$$\|u-z\|^2=(u_1-z_1)^2+\cdots+(u_l-z_l)^2+(z_{l+1}-z_{l+1})^2+\cdots+(z_m-z_m)^2=$$
$$=(u_1-z_1)^2+\cdots+(u_l-z_l)^2<\varepsilon^2.$$

Таким образом, построенное u содержится в $B_\varepsilon^m(z)$.

Возвращаясь к доказательству леммы, очевидным образом получаем, что вместе с точкой z' множество U' содержит и некоторую ее окрестность (шар $B_\varepsilon^l(z')$). Понятно, что $B_\varepsilon^l(z')\subset \operatorname{int}U'$, так как все точки шара являются внутренними точками множества U'. Таким образом, по определению множество $f(\operatorname{int}U)$ является открытым.

Доказательство леммы 2. Мы приведем множество, содержащееся в U и очевидно содержащее внутреннюю точку.

Для начала покажем, что мера Лебега любого подпространства $Q\subset R^m$, имеющего размерность $\dim Q=p<m$, равна нулю.

Пусть $e_1,\ldots,e_m$ – ортонормированный базис в R^m. Не ограничивая общности, будем считать, что $Q=L(e_1,\ldots,e_{m-1})$ – линейная оболочка, натянутая на первые p базисных векторов. Введем множества $R_n=\{z: z=(z_1,\ldots,z_m)\in R^m,\ |z_1|<n,\ldots,|z_m|<n\}$ для каждого $n\in N$. Тогда множество Q можно представить следующим образом: $Q=\bigcup_{i=1}^{\infty}Q_1$, где $Q_i=Q\cap R_i$. Заметим, что $Q_i\subset Q_{i+1}\ \forall i\in N$. По определению меры Лебега λ выполнено: $\lambda(Q_n)\le\lambda(R_n)=$

$= \overbrace{n * \cdots * n}^{p \text{ раз}} * \overbrace{0 * \cdots * 0}^{m-p \text{ раз}} = 0$ Тогда по свойству непрерывности меры получаем $\lambda(Q) = \lim_{n\to\infty} \lambda(Q_n) = \lim_{n\to\infty} 0 = 0$.

Нам известно, что множество U выпукло, замкнуто, ограничено и $P_1(U) = 1$. Ясно, что мера Лебега множества U положительна, так как иначе в силу абсолютной непрерывности меры P_1 относительно меры Лебега мы имели бы $P_1(U) = 0$.

Множество U содержит некоторую точку z_1. Также множество U содержит отличную от нее точку z_2, так как иначе мера Лебега множества U была бы равна нулю.

Так как множество U выпукло, то оно содержит и выпуклую оболочку множества $\{z_1, z_2\}$. Выпуклой оболочкой множества C называется пересечение всех выпуклых множеств, содержащих в себе множество C, то есть это есть наименьшее выпуклое множество, содержащее C. Выпуклую оболочку множества C будем обозначать $convC$. Для множества $\{z_1, z_2\}$ выпуклой оболочкой будет являться отрезок, соединяющий точки z_1 и z_2, обозначим этот отрезок через C_2.

Множество U содержит точки, отличные от точек множества C_2, так как в ином случае множество U содержалось бы в пространстве размерности, меньшей m и, по доказанному выше, имело бы меру 0. Обозначим одну из таких точек через z_3. В силу выпуклости множество U содержит множество $C_3 = conv\{C_2 \cup z_3\}$, которое является треугольником.

Продолжая рассуждать аналогичным образом, мы получим некоторое множество $C_{m+1} \subset U$, которое можно представить как многомерную пирамиду. Это множество не содержится ни в каком подпространстве размерности меньшей, чем размерность пространства R^m.

Множество C_{m+1}, очевидно, имеет внутреннюю точку.

Доказательство леммы 3. По лемме 2 в множестве U существует внутренняя точка, которую мы обозначим через z_{int}. Вместе с этой точкой множество U содержит некоторую окрестность

$B_\varepsilon(z_{\text{int}}) = \{z : z \in R^m,\ \|z - z_{\text{int}}\| < \varepsilon\}$ для некоторого $\varepsilon > 0$. В силу выпуклости множества U оно содержит и выпуклую оболочку множества $B_\varepsilon(z_{\text{int}}) \cup z'$, которую мы обозначим через $B_C = conv\{B_\varepsilon(z_{\text{int}}) \cup z'\}$.

Теперь, если мы возьмем последовательность точек $z_n = z' + \frac{1}{n}(z_{\text{int}} - z') = \frac{1}{n} z_{\text{int}} + \frac{n-1}{n} z'$, то она в силу выпуклости множества U будет состоять из точек множества U. Покажем, что все точки z_n являются внутренними. Мы докажем, что для z_n в множестве U содержится следующая окрестность точки: $B_{\varepsilon/n}(z_n) = \{z : z \in R^m,\ \|z - z_n\| < \varepsilon/n\}$. Возьмем некоторую точку $w \in B_{\varepsilon/n}(z_n)$, тогда $\|w - z_n\| = \left\| w - \frac{1}{n} z_{\text{int}} - \frac{n-1}{n} z' \right\| < \varepsilon/n$. Если мы рассмотрим точку $w' = z' + n(w - z')$, то выполнено

$$\|w' - z_{\text{int}}\| = \|z' + (w - z')n - z_{\text{int}}\| = n \left\| w - \frac{n-1}{n} z' - \frac{1}{n} z_{\text{int}} \right\| < n * \varepsilon / n = \varepsilon,$$

то есть точка w' принадлежит множеству $B_\varepsilon(z_{\text{int}})$. К тому же $\frac{1}{n} w' + \frac{n-1}{n} z' = \frac{1}{n} z' + (w - z') + \frac{n-1}{n} z' = w,$ поэтому точка w содержится в множестве $U,$ то есть во множестве U содержится окрестность точки z_n. Таким образом, мы в явном виде выписали последовательность внутренних точек множества $U,$ сходящуюся к точке z'.

Доказательство леммы 4.

Нам надо показать, что $[f(\text{int}U)] = U'$. Сделаем это, доказав включение в обе стороны.

Пусть $z \in [f(\text{int}U)]$, тогда существует последовательность $z_1, z_2, \ldots$ точек из $f(\text{int}U)$, сходящаяся к $z.$ А так как $f(\text{int}U) \subset U'$ и U' замкнуто, то $z \in U'$.

Пусть теперь $z' \in U'$. Возьмем точку $z \in U$ такую, что $f(z) = z'$. Если $z -$ внутренняя точка множества U, то очевидно, существует последовательность $z_1, z_2, \ldots$ внутренних точек множества U, сходящаяся к z. В противном случае существование такой последова-

тельности гарантирует лемма 3. Из леммы 1 следует, что точки $f(z_1), f(z_2),\ldots$ будут внутренними для множества U'. Из непрерывности оператора f следует, что последовательность $f(z_1), f(z_2),\ldots$ будет сходиться к $f(z) = z'$. То есть $z' \in \left[f(\mathrm{int}U) \right]$ и, следовательно, $U' \subset \left[f(\mathrm{int}U) \right]$.

Мы показали, что $f(\mathrm{int}U) \subset U'$ и $U' \subset \left[f(\mathrm{int}U) \right]$, поэтому $\left[f(\mathrm{int}U) \right] = U'$.

Доказательство леммы 5.

Здесь всюду $\lambda \in [0;1]$ – произвольное число.

Пусть $z_1, z_2 \in \mathrm{int}U$, тогда $\exists \varepsilon > 0$ такое, что $B_\varepsilon(z_1) \subset \mathrm{int}U$ и $B_\varepsilon(z_2) \subset \mathrm{int}U$. Как нетрудно видеть, тогда для $z = \lambda z_1 + (1-\lambda) z_2$ в силу выпуклости U выполнено $B_\varepsilon(z) \subset \mathrm{int}U$.

Пусть $z_1', z_2' \in f(\mathrm{int}U)$ и $\lambda \in [0;1]$.

Возьмем точки $z_1, z_2 \in \mathrm{int}U$ такие, что $f(z_1) = z_1'$ и $f(z_2) = z_2'$. Так как множество $\mathrm{int}U$ выпукло, то $\lambda z_1 + (1-\lambda) z_2 \in \mathrm{int}U$. Следовательно, $f(\lambda z_1 + (1-\lambda) z_2) = \lambda f(z_1) + (1-\lambda) f(z_2) \in f(\mathrm{int}U)$.

Выпуклость $\mathrm{int}U'$ доказывается аналогично выпуклости $\mathrm{int}U$.

Заключение

Вопрос о понижении размерности исследуемых данных часто встает перед исследователем. Нередко можно исследовать 2–3 фактора вместо исходных 20–30, получая при этом не менее точные результаты. При проведении кластеризации (выделении однородных групп) часто возникают сложности вычислительного характера. Хоть в наше время вопрос вычислительных мощностей не стоит так остро, как раньше, но все же лучше обойти эти сложности, используя аналитические технологии. Поэтому разумно в соответствующих ситуациях постараться перейти к данным меньшей размерности.

Теоретически для применения метода К-средних необходимо выполнение некоторых предположений о структуре данных. При понижении размерности мы в любом случае теряем часть информации и начинаем работать уже с другими данными. В статье показано, что если для исходных данных были выполнены необходимые предположения, то и для результата понижения размерности тоже будут выполнены необходимые предположения.

Примечания

[1] См.: *MacQueen J.* Some methods for classification and analysis of multiva-riate observations. Los Angeles: University of California, 1967.

[2] См.: Справочник по прикладной статистике / Под ред. Э. Ллойда, У. Ледермана. М.: Финансы и статистика, 1989. Т. 2.

[3] См.: *Тыртышников Е.Е.* Матричный анализ и линейная алгебра. М.: ИВМ РАН, 2004–2005.

Ю.В. Козлова

ГЕНЕРАТОР ТЕСТОВЫХ ПРИМЕРОВ ДЛЯ РАЗЛИЧНЫХ ВАРИАНТОВ ДСМ-МЕТОДА

Описывается программа, порождающая тестовые данные для демонстрации особенностей различных вариантов ДСМ-метода. Рассматриваются примеры работы экспериментальной ДСМ-системы с данными, порожденными этой программой.

Ключевые слова: ДСМ-метод, ДСМ-система, варианты ДСМ-метода, тестирование, тестовые данные.

ДСМ-метод автоматического порождения гипотез был предложен В.К. Финном в конце 70-х гг. XX в. Название метода образовано из инициалов Джона Стюарта Милля[1], правила индуктивной логики которого и лежат в основе ДСМ-метода. Д.С. Милль сформулировал правила правдоподобных рассуждений, названные им: «метод сходства», «метод различия», «объединенный метод сходства и различия», «метод сопутствующих изменений» и «метод остатков».

ДСМ-метод предназначен для обнаружения закономерностей в предметной области. А именно с помощью ДСМ-метода устанавливаются причинно-следственные связи между структурой объектов предметной области и их свойствами. Интеллектуальные системы, основанные на ДСМ-методе, могут применяться в разных областях науки, где знания слабо формализованы, а данные хорошо структурированы или допускают структуризацию. Систему интеллектуального анализа данных, основанную на ДСМ-методе, называют ДСМ-системой.

ДСМ-метод порождает гипотезы о связи между структурой объекта и его свойствами, а также гипотезы о наличии или отсут-

ствии свойств у тех объектов, о которых нет экспериментальных данных.

Данные, с которыми работает ДСМ-метод, имеют вид информационных массивов, которые будем называть «база фактов» и «база знаний».

База фактов ДСМ-системы содержит сведения о структуре объектов и о наличии или отсутствии у них целевых свойств. *База знаний* содержит правила ДСМ-метода, систему вспомогательных вычислительных процедур и сведения о причинах наличия или отсутствия у объектов целевых свойств. Последние сведения также необходимо считать знаниями, так как они выражают общую закономерность. Если отвлечься от возможных мешающих факторов, то предложение «фрагмент структуры *s* является возможной причиной целевого свойства *p*» можно интерпретировать как общее утверждение «каждый объект, содержащий фрагмент *s*, обладает свойством *p*».

В ДСМ-логике существует четыре типа внутренних истинностных значений:

+1 – эмпирическая истина,

–1 – эмпирическая ложь,

0 – эмпирическое противоречие,

τ – неопределенность.

Неформально эти типы истинностных значений можно интерпретировать следующим образом.

Если гипотезе о том, что структурная особенность *s* является причиной наличия свойства *p*, присваивается истинностное значение типа:

+1, то *s* является причиной наличия свойства *p* (или «+» - причиной для *p*),

–1, то *s* является причиной отсутствия свойства *p* (или «–» - причиной для *p*),

0, то *s* может являться как причиной наличия свойства *p*, так и причиной его отсутствия,

τ, то отсутствуют и данные, подтверждающие гипотезу, и данные, опровергающие гипотезу.

Если гипотезе о том, что объект *o* обладает свойством *p*, присваивается истинностное значение типа:

+1, то это утверждение эмпирически истинно – *o* обладает свойством *p* (*o* – «+» - пример для *p*),

–1, то это утверждение эмпирически ложно (объект *o* не обладает свойством *p*, *o* – «–» - пример для *p*),

0, то есть как аргументы за то, что объект *o* обладает свойством *p*, так и аргументы за то, что объект *o* не обладает свойством *p* (*o* – 0 - пример для *p*),

τ, то данные о том, обладает ли объект *o* свойством *p*, отсутствуют (*o* – недоопределенный пример, τ - пример для *p*).

При работе ДСМ-системы учитываются сведения о структуре объектов, анализируются и модифицируются сведения о наличии или отсутствии свойств и возможных причинах наличия или отсутствия свойств.

Данные для ДСМ-системы можно представить в виде матриц, где индексами строк и столбцов являются названия (или идентификаторы) объектов, свойств или структурных элементов (атомов), а элементами – типы истинностных значений.

Так, в матрице, соответствующей системе сведений о наличии или отсутствии у объектов целевых свойств (матрица *F*), строки индексируются элементами из множества объектов, столбцы – элементами из множества целевых свойств, а на их пересечении стоит один из типов истинностных значений.

В матрице, представляющей систему сведений о возможных причинах наличия или отсутствия целевых свойств (матрица *H*), строки индексируются элементами из множества структурных особенностей, а столбцы – элементами из множества целевых свойств.

В матрице *S*, соответствующей системе сведений о структуре объектов, индексами строк являются объекты, индексами столбцов – структурные фрагменты, а элементы матрицы – **t** или **f** (соответственно, истина или ложь из класса внешних истинностных значений) в зависимости от того, включает ли объект *o* структурный фрагмент *s*.

В процессе работы ДСМ-метод доопределяет матрицу *F* и матрицу *H* с помощью порождаемых гипотез, то есть происходит замена неопределенности **t** на определенные значения.

F	p_1	⋯	p_n
o_1	+1	⋯	0
⋮	⋮		⋮
o_m	–1	⋯	τ

Матрица *F*

H	p_1	⋯	p_n
s_1	+1	⋯	0
⋮	⋮		⋮
s_m	–1	⋯	τ

Матрица *H*

S	s_1	⋯	s_n
o_1	**t**	⋯	**f**
⋮	⋮		⋮
o_m	**f**	⋯	**t**

Матрица *S*

Рис. 1. Представление базы фактов и базы знаний в виде матриц

Разновидности ДСМ-метода рассмотрены в работах Н.И. Мельникова[2], С.М. Гусаковой[3], М.А. Михеенковой и В.К. Финна[4]. На сегодняшний день выделяют следующие варианты ДСМ-метода:

1) простой ДСМ-метод строит гипотезы в предположении о том, что могут существовать как причины наличия, так и причины отсутствия свойства у объекта («+»- и «–»-причины), при этом не учитывается возможное окружение причины / антипричины, то есть контекст;
2) в ДСМ-методе с тормозами предполагается, что причины или антипричины свойства могут блокироваться другими элементами в структуре объекта (тормоза причины или тормоза антипричины); в свою очередь ДСМ-метод с тормозами имеет следующие варианты:
   a) в обобщенном ДСМ-методе тормоза могут быть как для причины наличия свойства, так и для причины отсутствия свойства (то есть для антипричины),
   b) в несимметричном ДСМ-методе поиск тормозов идет только для причин наличия свойства, считается, что причиной отсутствия проявления свойства может быть не наличие антипричины, а лишь отсутствие вхождения какой-либо из причин;
3) также выделяют прямой и обратный ДСМ-методы:
   a) в прямом ДСМ-методе происходит поиск возможных причин при известных следствиях (целевых свойствах),
   b) в обратном ДСМ-методе ищутся возможные следствия при известных причинах, которые играют в обратном ДСМ-методе роль, аналогичную роли целевых свойств в прямом ДСМ-методе.

В статье рассматривается программа «TestJSM!» для порождения тестовых примеров, демонстрирующих возможности разных вариантов ДСМ-метода. В частности, эта программа может быть использована для проверки вариантов ДСМ-метода, а также для сравнения выразительной силы ДСМ-стратегий[5].

Пользователь имеет возможность подробно описать систему требований к тестовым данным и сохранить ее в файле. Ранее сохраненные требования можно загружать и редактировать. Система требований сохраняется в формате XML в соответствии со своей объектной моделью. Главной функцией представляемой программы является генерация тестовых примеров, удовлетворяющих обсуждаемой ниже системе требований.

Объектная модель системы требований

Первоначально представляемая в данной работе программа предназначалась для генерации учебных задач по ДСМ-методу. Поэтому пользователь (преподаватель) имел большие возможности по настройке системы требований к тестовым примерам. Впоследствии главной функцией программы стала генерация тестовых примеров для ДСМ-систем. В настоящее время предлагаемая программа обладает следующими возможностями:

– порождать данные для проверки и сравнения эффективности разных ДСМ-стратегий;
– использоваться для тестирования разрабатываемых ДСМ-систем;
– использоваться как генератор задач для студентов, изучающих ДСМ-метод.

Настройка параметров требований осуществляется с помощью мастера, который последовательно предлагает пользователю:

– выбор ДСМ-стратегии,
– ввод желаемых причин, свойств, тормозов,
– определение ограничений на объекты, порождающие причины и тормоза.

Объектная модель включает коллекции и объекты. На диаграмме объектной модели названия коллекций находятся в прямоугольниках с толстой рамкой, а названия объектов – в прямоугольниках с тонкой рамкой (рис. 2). В структуру объектной модели входят следующие компоненты.

1. Коллекция «Набор систем требований» – самый верхний компонент в иерархии объектной модели, который включает в себя все остальные компоненты. Каждый объект «Система требований» данной коллекции включает:

a) коллекцию «Свойства»,
b) коллекцию «Шаги работы ДСМ-метода».

2. Коллекция «Свойства» – названия свойств, для которых указываются причины наличия/отсутствия свойств или информация о том, что данных о связи свойств и фрагмента нет; причем предполагается, что соответствующая гипотеза выдвигается не раньше чем на n-м шаге работы ДСМ-метода ($n \geqslant 1$).

3. Коллекция «Шаги работы ДСМ-метода». Каждый объект «Шаг работы ДСМ-метода» данной коллекции включает в себя коллекцию «Причины». Предполагается, что на n-м шаге работы ДСМ-метода во время процедуры индукции выдвигается гипотеза о причине / антипричине целевых свойств.

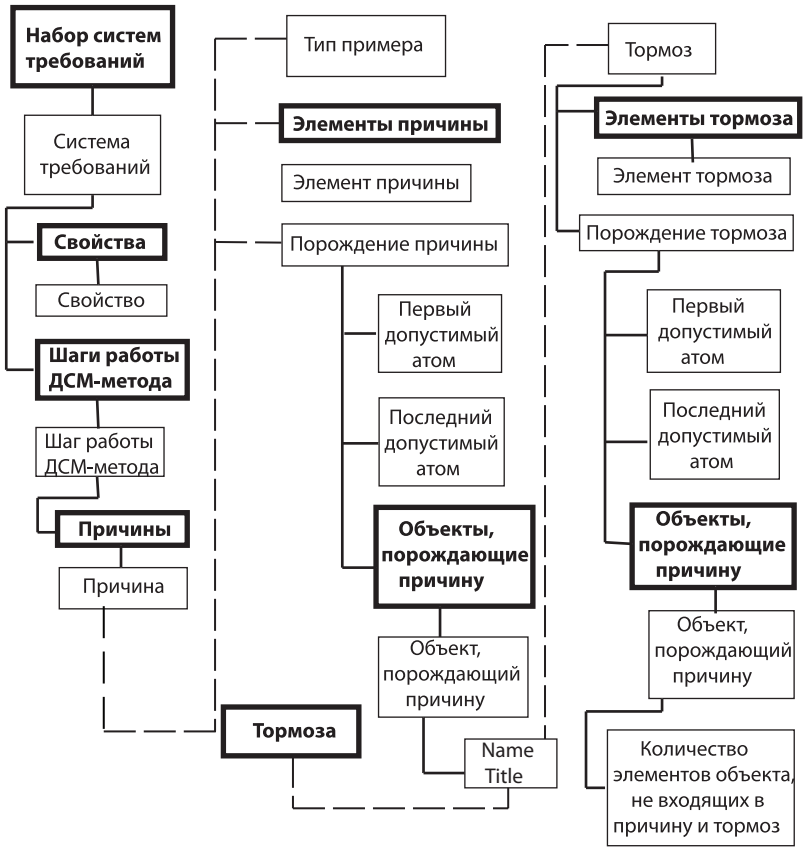


Рис. 2. Объектная модель системы требований к тестовым примерам

4. Коллекция «Причины». Каждый объект «Причина» данной коллекции включает в себя следующие компоненты:

a) объект «Тип примера» (означает тип истинностного значения получаемого примера, то есть +1, –1 или τ),

b) коллекцию «Элементы причины»,

c) коллекцию «Тормоза»,

d) объект «Порождение причины».

5. Коллекция «Элементы причины» включает в себя элементы структуры причины.

6. Объект «Порождение причины» содержит информацию об объектах, при пересечении которых порождается причина нали-

чия / отсутствия свойств. Этот объект включает следующие компоненты:

a) объекты «Первый допустимый атом» и «Последний допустимый атом» (элементы объектов, не входящие в причину, порождаются генератором автоматически, поэтому у пользователя запрашивается диапазон атомов, из которого берутся оставшиеся элементы, данные компоненты и означают границы диапазона),

b) коллекцию «Объекты, порождающие причину»: для каждого объекта пользователь заполняет свойство «Количество элементов, не входящих в причину».

7. Коллекция «Тормоза» содержит информацию о возможных тормозах данной причины / антипричины свойств (при этом если нет данных о связи объекта и свойства, то информация о тормозах не вводится). Данная информация запрашивается у пользователя при генерации примеров для стратегий ДСМ-метода с тормозами. Каждый объект «Тормоз» данной коллекции включает в себя:

a) коллекцию «Элементы тормоза»,

b) объект «Порождение тормоза».

8. Коллекция «Элементы тормоза» включает в себя элементы структуры тормоза.

9. Объект «Порождение тормоза» содержит информацию об объектах, при пересечении которых порождается причина наличия/отсутствия свойств и ее тормоз. Этот объект содержит следующие компоненты:

a) объекты «Первый допустимый атом» и «Последний допустимый атом» (элементы объектов, не входящие в причину и тормоз, порождаются генератором автоматически; поэтому у пользователя запрашивается диапазон элементов, из которого берутся оставшиеся элементы, данные компоненты и означают границы диапазона),

b) коллекцию «Объекты, порождающие тормоз». Для каждого объекта пользователь заполняет свойство «Количество элементов, не входящих в причину и тормоз».

На рис. 2 представлена структура объектной модели системы требований.

Описание работы генератора тестовых примеров «TestJSM!»

Основными функциями генератора примеров для ДСМ-метода являются:

– ввод новой системы требований к примерам в соответствии с объектной моделью;

– загрузка ранее сохраненной системы требований из файла формата XML;
– порождение примеров в виде таблицы «объект–свойство»; получаемая таблица является комбинацией матрицы *F* и матрицы *S*; то есть в этой матрице строки индексируются элементами из множества объектов, которые представлены в виде множеств, состоящих из структурных элементов, а столбцы – элементами из множества целевых свойств, на пересечении строк и столбцов стоит один из типов истинностных значений;
– сохранение примеров в файлах форматов CSV и XLS;
– многократное порождение разных примеров для одной и той же системы требований.

Опишем главную функцию системы – генерацию тестовых примеров.

Приведем систему правил, с помощью которых порождаются тестовые данные для простой ДСМ-стратегии. Для этого введем следующие предикатные переменные:

1) start_jsm (ListEl, First, Last, [Elnum | Rest], Fin) – 5-арная предикатная переменная, описывающая отношение между параметрами, задаваемыми пользователем (ListEl, First, Last, [Elnum | Rest]), и конечным результатом работы системы правил (Fin):

a) ListEl – список элементов рассматриваемой причины,

b) First и Last – начало и конец диапазона атомов, образующих объекты, пересечением которых является рассматриваемая причина,

c) Fin – список объектов, пересечением которых является упомянутая выше причина,

d) [Elnum | Rest] – список, который для каждого объекта содержит количество элементов, не входящих в рассматриваемую причину, указывается первый элемент списка (Elnum) и оставшиеся элементы (Rest);

2) final_list (ListEl, First, Last, Elnum, List) – 5-арная предикатная переменная, имеющая смысл, аналогичный предыдущей предикатной переменной:

a) ListEl, First, Last, Elnum понимаются так же, как и в предыдущем предикате,

b) List – список элементов одного из объектов, содержащих причину;

3) list (ListEl, First, Last, Elnum, List1) – 5-арная предикатная переменная, которая описывает отношение между данными, задаваемыми пользователем (ListEl, First, Last, Elnum), и порождаемым с помощью генератора псевдослучайных чисел списком элементов, не входящих в причину (List1):

a) ListEl, First, Last, Elnum понимаются так же, как и в первом предикате,

b) List1 – список элементов, не входящих в причину;

4) append (ListEl, List1, Res) – тернарная предикатная переменная:

a) ListEl – список элементов рассматриваемой причины,

b) List1 – список элементов, не входящих в причину,

c) Res – список, являющийся объединением списков ListEl и List1;

5) casual (First, Last, El) – тернарная предикатная переменная:

a) First и Last – начало и конец диапазона атомов, образующих объекты, пересечением которых является рассматриваемая причина,

b) El – случайный элемент из диапазона [First, Last];

6) member (El, ListEl) – бинарная предикатная переменная, означающая, что элемент El входит в список ListEl; параметры El и ListEl описаны выше;

7) rest_els (Els) – унарная предикатная переменная, используемая для сохранения списков элементов, не входящих в причину: Els – список списков элементов, не входящих в причину (для каждого объекта формируется свой список элементов, не входящих в причину);

8) member2 (El, Els) – бинарная предикатная переменная, означающая, что элемент El входит в один из элементов списка Els; параметры El и Els описаны выше.

Генерация тестовых примеров для простого ДСМ-метода может быть записана как следующая система правил:

Пополнение списка объектов, содержащих причину,
start_jsm (ListEl, First, Last, [Elnum | Rest], Fin) ←
final_list (ListEl, First, Last, Elnum, List),
start_jsm (ListEl, First, Last, Rest, [List | Fin]).

Формирование списка элементов одного объекта
final_list (ListEl, First, Last, Elnum, Res) ←
list (ListEl, First, Last, Elnum, List1),
append (ListEl, List1, Res).

Формирование списка элементов, не входящих в причину (для каждого объекта формируется свой список),
list (ListEl, First, Last, Elnum, RestList) ←
Elnum > 0,
casual (First, Last, El),
  member (El, ListEl),
  member (El, RestList),
rest_els (Els),
  member2 (El, Els),

Elnum1 = Elnum –1,
list (ListEl, First, Last, Elnum1, [El | RestList]).

Опишем систему правил, с помощью которой порождаются тестовые данные для ДСМ-стратегии с тормозами. В данной системе правил используются те же предикатные переменные, которые использовались при генерации примеров для простого ДСМ-метода. Также вводятся некоторые другие предикатные переменные:

1) append_el_torm (ListEl, [Torm | List_torm], List_el_torm) – тернарная предикатная переменная:

a) ListEl – список элементов рассматриваемой причины,

b) [Torm | List_torm] – список списков элементов, входящих в тормоза рассматриваемой причины, указывается первый элемент списка (Torm) и оставшиеся элементы (List_torm),

c) List_el_torm – список списков элементов, являющихся объединением списка ListEl и каждого из списков [Torm | List_torm];

2) append_lists (ListEl, Torm, List) – бинарная предикатная переменная:

a) параметр ListEl описан выше,

b) Torm – список элементов тормоза рассматриваемой причины,

c) List – список, являющийся объединением списков ListEl и Torm;

3) tormoza (List_torm) – унарная предикатная переменная, используемая для сохранения списка элементов, входящих в тормоза причины; List_torm – список списков элементов, входящих в тормоза причины (для каждой причины формируется свой список элементов, входящих в тормоза);

4) check_for_tormoza (List_torm, List) – бинарная предикатная переменная, в которой происходит проверка, что ни один из тормозов списка списков List_torm не входит в список List:

a) параметр List_torm описан выше,

b) List – список элементов, не входящих в причину;

5) len (El, Len) – бинарная предикатная переменная:

a) El – список элементов,

b) Len – длина списка;

6) sublist (H, Fin_list) – бинарная предикатная переменная, в которой происходит проверка, что все элементы тормоза H не входят в список элементов Fin_list.

Генерация тестовых примеров для ДСМ-метода с тормозами может быть записана как следующая система правил:

Формирование списков, содержащих элементы причины и один из ее тормозов

append_el_torm (ListEl, [Torm | List_torm], List_el_torm) ←
append_lists (ListEl, Torm, List),
append_el_torm (ListEl, List_torm, [List | List_el_torm]).

Пополнение списка объектов, содержащих причину (или же причину и один из ее тормозов),

```
start_jsm (ListEl, First, Last, [Elnum | Rest], Fin) ←
final_list (ListEl, First, Last, Elnum, List),
start_jsm (ListEl, First, Last, Rest, [List | Fin]).
```

Формирование списка элементов одного объекта

```
final_list (ListEl, First, Last, Elnum, Res) ←
list (ListEl, First, Last, Elnum, List1),
append (ListEl, List1, Res),
tormoza (List_torm),
check_for_tormoza (List_torm, List1).
```

Формирование списка элементов, не входящих в причину (для каждого объекта формируется свой список),

```
list (ListEl, First, Last, Elnum, RestList) ←
Elnum > 0,
casual (First, Last, El),
  member (El, ListEl),
  member (El, RestList),
rest_els (Els),
  member2 (El, Els),
Elnum1 = Elnum –1,
list (ListEl, First, Last, Elnum1, [El | RestList]).
```

Проверка того, что ни один из тормозов списка не входит в список элементов, не входящих в причину,

```
check_for_tormoza ([H|T], Fin_list) ←
len (H, Len1),
      len (Fin_list, Len2),
      Len2 >= Len1,
      sublist (H, Fin_list),
      check_for_tormoza (T, Fin_list).
```

Работа экспериментальной ДСМ-системы с тестовыми данными, порожденными генератором «TestJSM!»

Рассматриваемая ниже ДСМ-система была разработана О.А. Сафроновой[6] на языке Visual Prolog на основе системы правил, предложенных[7] и развитых Д.В. Виноградовым[8]. В этой экспериментальной ДСМ-системе в отличие от профессиональных ДСМ-систем реализованы все основные разновидности ДСМ-стратегий.

Простой ДСМ-метод без итераций и простой ДСМ-метод с итерациями

Итак, сравним работу простого ДСМ-метода без использования итераций и работу простого ДСМ-метода, использующего итерации. Оба метода прямые с запретом на контрпример.

С помощью генератора «TestJSM!» было порождено 10 тестовых примеров с одним целевым свойством для простого ДСМ-метода с итерациями. После этого полученные примеры были последовательно поданы на вход ДСМ-системы для простого ДСМ-метода без итераций и для простого ДСМ-метода с итерациями.

Результаты работы простого ДСМ-метода без итераций и с итерациями сравнивались по следующим параметрам:

- количество гипотез, порожденных на этапе индукции,
- количество гипотез, порожденных на этапе аналогии (что соответствует и количеству доопределенных примеров),
- количество «+»-примеров среди доопределенных,
- количество «–»-примеров среди доопределенных,
- количество «0»-примеров среди доопределенных.

На следующих диаграммах представлено сравнение методов по первым двум параметрам (рис. 3–4).

На диаграммах хорошо видно, что при использовании итераций в простом ДСМ-методе количество порождаемых гипотез как на этапе индукции, так и на этапе аналогии значительно больше, чем количество гипотез, полученных в процессе работы простого ДСМ-метода без итераций.

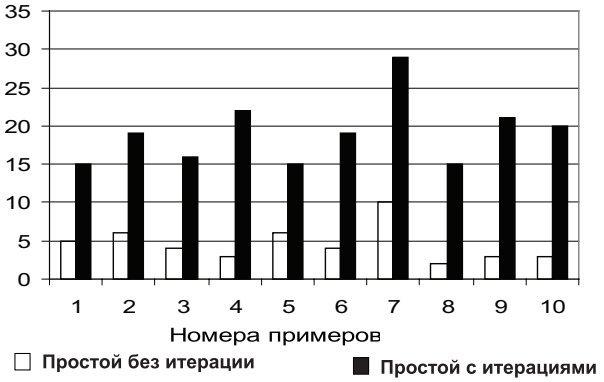


Рис. 3. Сравнение простого ДСМ-метода с итерациями и без итераций. Количество гипотез на этапе индукции

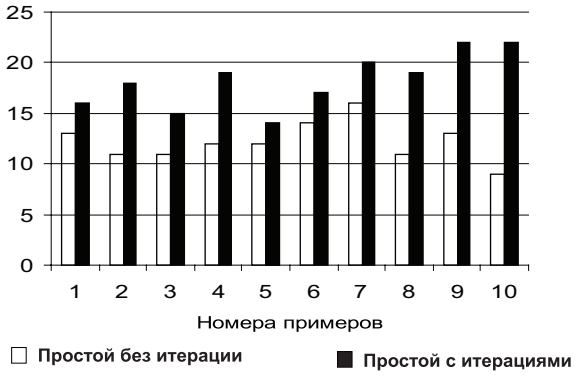


Рис. 4. Сравнение простого ДСМ-метода с итерациями и без итераций. Количество гипотез на этапе аналогии

Следует заметить, что количество итераций не везде было одинаковым. Так, для примеров 1–5 было сделано 3 шага работы ДСМ-метода, для примеров 6–8 – 4 шага и для примеров 9–10 – 5 шагов. По диаграммам видно, что чем больше ДСМ-метод делает шагов, тем больше порождается гипотез.

На следующих диаграммах покажем количество доопределенных «+»-, «–»- и «0»-примеров в процессе работы ДСМ-метода (рис. 5–10). Можно заметить, что больше всего доопределено «0»-примеров, что объясняется спецификой простого ДСМ-метода, в результате работы которого получается много «0»-примеров в силу его бесконтекстности.

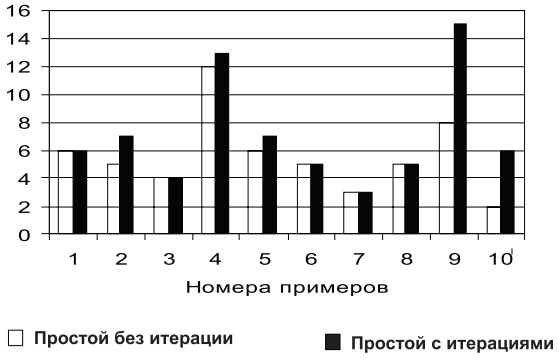


Рис. 5. Сравнение простого ДСМ-метода с итерациями и без итераций. Количество доопределенных «+»-примеров

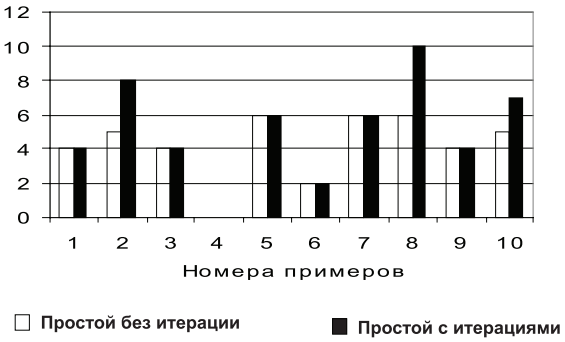


Рис. 6. Сравнение простого ДСМ-метода с итерациями и без итераций. Количество доопределенных «–»-примеров

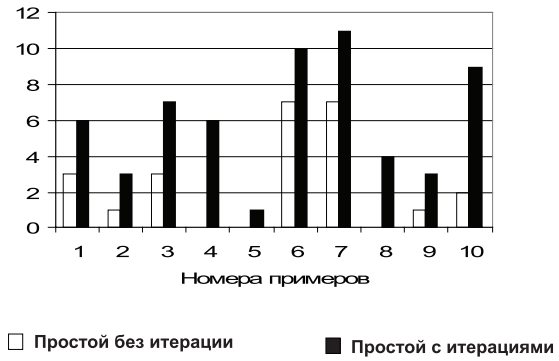


Рис. 7. Сравнение простого ДСМ-метода с итерациями и без итераций. Количество доопределенных «0»-примеров

Рис. 8. Сравнение простого ДСМ-метода и обобщенного ДСМ-метода. Количество доопределенных «+»-примеров

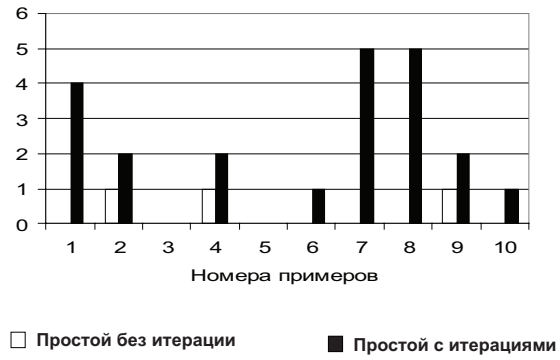


Рис. 9. Сравнение простого ДСМ-метода и обобщенного ДСМ-метода. Количество доопределенных «–»-примеров

Рис. 10. Сравнение простого ДСМ-метода и обобщенного ДСМ-метода. Количество доопределенных «0»-примеров

Простой ДСМ-метод без итераций и обобщенный ДСМ-метод

Сравним результаты работы простого ДСМ-метода с запретом на контрпример, не использующего итерации, и результаты работы обобщенного ДСМ-метода без итераций. Оба метода прямые.

С помощью генератора «TestJSM!» было порождено 10 тестовых примеров с одним целевым свойством для обобщенного ДСМ-метода. После этого примеры были поданы на вход ДСМ-системы

для простого ДСМ-метода без использования итераций и для обобщенного ДСМ-метода без итераций.

Результаты работы методов сравнивались по следующим параметрам:

– количество «+»-примеров среди доопределенных,
– количество «–»-примеров среди доопределенных,
– количество «0»-примеров среди доопределенных.

Сравнение по этим параметрам представлено на диаграммах 7–10.

Из сравнения результатов работы простого ДСМ-метода и обобщенного видны особенности каждой из стратегий, а именно, что обобщенный ДСМ-метод порождает больше положительных и отрицательных гипотез и вообще не порождает «0»-гипотез, а простой порождает достаточно большое количество противоречивых гипотез.

Итак, был продемонстрирован образец применения тестовых примеров, получаемых генератором «TestJSM!» в качестве входных данных для ДСМ-системы. Было показано, что тестовые примеры могут использоваться для тестирования работы ДСМ-системы, а также для сравнения разных вариантов ДСМ-метода. На основании сравнения можно выдвинуть определенные гипотезы о работе того или иного метода, но для этого нужна, без сомнения, более объемная статистика.

Примечания

1 См.: *Милль Д.С.* Система логики силлогистической и индуктивной. М.: Книжное дело, 1900.

2 *Мельников Н.И.* Решатель интеллектуальных задач типа ДСМ // НТИ. Сер. 2. 2000. № 4. С. 16–18.

3 *Гусакова С.М., Михеенкова М.А., Финн В.К.* О логических средствах автоматизированного анализа мнений // НТИ. Сер. 2. 2001. № 5. С. 4–24.

4 См.: *Финн В.К.* Правдоподобные рассуждения в интеллектуальных системах типа ДСМ // Итоги науки и техники. Сер. «Информатика». 1991. Т. 15: Интеллектуальные информационные системы.

5 Будем говорить, что выразительная сила стратегии S не больше, чем выразительная сила стратегии S', если все гипотезы, которые можно породить с помощью стратегии S, можно породить и с помощью стратегии S'. Можно сравнивать выразительную силу стратегий и для каких-либо конкретных разновидностей гипотез, например только для гипотез первого (второго) рода, только для положительных (отрицательных) гипотез и т. п.

6 См.: *Efimova E., Safronova O., Vinogradov D.* A Prototype of JSM-system in Visual Prolog // Proceedings of the First Visual Prolog Applications and Language Conference. Prolog Development Center. 2006.

7 См.: *Виноградов Д.В.* Логические программы для квазиаксиоматических теорий // НТИ. Сер. 2. 1999. № 1–2.

8 См.: *Виноградов Д.В.* Корректные логические программы для правдоподобных рассуждений // НТИ. Сер. 2. 2001. № 5.

М.А. Михеенкова

О ФОРМАЛИЗОВАННЫХ ЭВРИСТИКАХ КАЧЕСТВЕННОГО АНАЛИЗА СОЦИОЛОГИЧЕСКИХ ДАННЫХ*

В работе предлагается рассмотрение двух подходов к формализации эвристической схемы «сходство – аналогия – абдукция», которая применяется для качественного анализа социологических данных – автоматизированного извлечения зависимостей из имеющихся фактов. Один подход при реализации сходства опирается на аппарат булевой алгебры. Второй реализован в интеллектуальных системах, использующих ДСМ-рассуждения – класс когнитивных правдоподобных рассуждений.

Ключевые слова: формализованный качественный анализ, ДСМ-метод, автоматическое порождение гипотез, интеллектуальная система, правдоподобные рассуждения, булева алгебра.

Повсеместная распространенность статистических средств анализа эмпирических социологических данных не только не препятствует, но, возможно, и способствует осознанию недостаточности (а порой и неадекватности) их использования. Классическое выражение этой точки зрения представлено в известной работе П.А. Сорокина «Квантофрения»[1], где источником псевдоквантификации автору видится неадекватность используемого метода исследуемой социальной реальности. Особенно зримо эта неадекватность проявляется при анализе качественных (не количественных) социологических данных, что привело к доминированию другой крайности – отношения к стратегиям качественного анализа как принципиально неформализуемым. В самом общем виде такой анализ определяется как совокупность творческих эвристик общения исследователя и респондента, субъективно интерпретирующего социальные явления и процессы. Далее полученный эмпирический материал анализируется и обобщается иссле-

* Работа выполнена при поддержке РГНР (проект № 08-03-00145а).

дователем на основе его опыта, интуиции, знаний и т. п.[2] Субъективный характер результатов такого рода анализа заставляет говорить о назревшей необходимости развития формальных методов качественного анализа данных[3].

Одна из наиболее развитых и авторитетных методологий качественного анализа социологических данных – *обоснованная теория* (grounded theory)[4] – характеризует качественное исследование как построение теории на основе эмпирических фактов с использованием индуктивного анализа, причем сбор информации происходит вплоть до насыщения выборки, когда новые индикаторы, категории и т. п. (термины обоснованной теории) перестают формироваться. Такое понимание стратегии качественного анализа в сочетании с использованием формальных средств совпадает с современным представлением относительно основной задачи интеллектуального анализа данных – автоматического извлечения интерпретируемых зависимостей между различными факторами, неявно содержащимися в массивах данных[5].

Необходимость извлечения именно причинных зависимостей из эмпирических данных, принципиальная ограниченность исходной базы фактов, упоминавшийся выше тезис о необходимости выбора адекватных социальной реальности средств анализа – все это способствовало развитию нестатистических методов формализованного качественного анализа социологических данных[6]. К их числу относится, к примеру, так называемый качественный сравнительный анализ[7] (Qualitative Comparative Analysis, далее – QCA), использующий аппарат булевой алгебры, а в более поздних вариантах – в сочетании с аппаратом нечетких множеств[8]. В отечественных исследованиях одним из достаточно развитых подходов такого рода является ДСМ-метод автоматического порождения гипотез[9] (далее – ДСМ-метод АПГ или ДСМ-метод) в его варианте для анализа социологических данных[10], реализованный в интеллектуальных системах (далее – ИС) типа ДСМ[11].

ДСМ-метод представляет собой специальный класс рассуждений (ДСМ-рассуждения), реализующий синтез познавательных процедур – эмпирической индукции (формальных расширений и уточнений индуктивных методов английского философа и логика Д.С. Милля, в честь которого и назван метод), структурной аналогии и абдуктивного рассуждения Ч.С. Пирса как средства принятия гипотез на основе объяснения начальных данных[12]. Средством формализации ДСМ-рассуждений являются бесконечнозначные логики степеней правдоподобия порождаемых гипотез.

Основной задачей рассматриваемого качественного анализа данных является исследование типа каузальности «структура –

эффект» в соответствии с принципом структурализма: «сходство фактов влечет наличие (отсутствие) изучаемого эффекта и его повторяемость». Таким образом, (нестатистическое) сходство фактов, имеющих определенную структуру, служит источником детерминации явлений и может определяться на основе как логико-алгебраического (QCA), так и формально-индуктивного подходов (ДСМ-метод АПГ). Дополнение логико-алгебраических процедур поиска сходства QCA адекватными процедурами вывода по аналогии и абдуктивного объяснения позволило говорить о реализации общей эвристической схемы «сходство – аналогия – абдукция»[13]. Особенности такой реализации в обоих подходах и сравнение полученных результатов и составляют предмет настоящей работы.

Определим, прежде всего, круг задач, для решения которых предназначены рассматриваемые эвристики. В соответствии с представлением М. Вебера о необходимости развития в социологии каузального объяснения процесса действия, его направленности и последствий[14] мы можем говорить о следующих задачах формализованного качественного анализа социологических данных:

– исследование индивидуального поведения, порождение детерминант поведения и типологизация социума на их основе;

– анализ и прогнозирование мнений респондентов как варианта поведения;

– выяснение влияния ситуации на поведение индивидуума;

– анализ рациональности мнений (в том числе степени рациональности мнений данной социальной общности).

В настоящей работе мы сосредоточимся на задаче анализа мнений, отсылая читателя для получения общего представления об указанной проблематике к другим работам[15].

Охарактеризуем онтологические допущения относительно особенностей исследуемой предметной области. Мы рассматриваем социальные явления (к примеру, индивидуальное поведение в социуме или мнение респондента) как причинно обусловленные, понимая при этом под причинной обусловленностью предрасположенность (согласно К. Попперу) к совершению поведенческих актов (действий, установок, мнений). Причем предрасположенность эта реализуется при отсутствии противодействующих влияний (как внутренних – личностных, так и внешних – ситуационных).

Принципиальным для предлагаемых эвристик является предположение о наличии как позитивных, так и негативных фактов, т. е. примеров наличия или отсутствия исследуемого явления, вызванного позитивными (+) и негативными (–) причинами (наиболее существенными и устойчивыми влияниями) соответственно.

Выполнение этого условия позволяет автоматически порождать фальсификаторы порожденных гипотез и может рассматриваться как основание для абдуктивного принятия индуктивных гипотез о причинах.

Из принципа структурализма вытекает необходимость предварительной (алгебраической) формализации сходства объектов и их свойств, на основе которого формируются гипотезы о причинах. Для рассматриваемого варианта формализованного качественного анализа социологических данных с использованием указанных эвристик основой представления знаний о социальных субъектах (как индивидах, так и социальных общностях) является так называемый постулат поведения. Пусть имеются три множества характеристик, входящих в описание субъекта поведения: признаки, представляющие социальный характер субъекта (*SC*); индивидуальные черты личности (*IP*); биографические данные (*BD*). Поведение *B* субъекта C определяется подмножеством характеристик $Det \subseteq C$ таким, что $Det = Det_1 \cup Det_2 \cup Det_3$, где $(Det_1 \subseteq (SC)) \& (Det_2 \subseteq (IP)) \& (Det_3 \subseteq (BD))$, причем хотя бы одно $Det_i \neq \varnothing$, $i = 1, 2, 3$. Таким образом, индивидуальные характеристики социального субъекта являются информативным основанием для порождения детерминант социального поведения и, соответственно, предсказания возможного поведения.

Описанные допущения – наличие исходных позитивных (+) и негативных (−) примеров (эмпирических фактов) изучаемых эффектов поведения, выявление (±) причин (существенных влияний) проявления этих эффектов на основании формализованного отношения сходства между фактами – формируют базис для адекватного использования формализованных эвристик «сходство – аналогия – абдукция».

В ДСМ-методе автоматического порождения гипотез указанная схема представляется в виде синтеза индукции, аналогии и абдукции, *InAnAb* и в общем виде может быть описана следующим образом.

В исходном состоянии базы фактов (далее – БФ) утверждения «субъект X обладает эффектом поведения Y» (X – структурированное описание субъекта, например в соответствии с постулатом поведения, Y – переменная для представления действий, установок и мнений) представлены предикатом $X \Rightarrow_1 Y$. На основе индуктивного анализа представленных этим предикатом примеров порождаются предикаты причинности $V \Rightarrow_2 W$ (прямой) или $W_3 \Leftarrow V$

(обратный), интерпретирующиеся как «подмножество характеристик V есть причина эффекта поведения W» и «эффект поведения W есть следствие подмножества характеристик V» соответственно. Полученные гипотезы о причинах используются в выводе по аналогии для расширения и уточнения представленного в начальном состоянии базы фактов отношения $\Rightarrow_1^*$ (например, предсказания возможных мнений). Цикл «индукция – аналогия» повторяется до стабилизации множества гипотез, полученных как с использованием правил правдоподобного вывода 1-го рода (далее – п.п.в.-1) – индукции (гипотезы о причинах), так и с использованием правил правдоподобного вывода 2-го рода (далее – п.п.в.-2) – аналогий (предсказательные гипотезы). Абдуктивное рассуждение – процедура объяснения начального состояния БФ полученными гипотезами – завершает ДСМ-рассуждение.

Опишем теперь кратко необходимые формальные средства ДСМ-метода АПГ[16], использующиеся для анализа мнений в соответствии с описанной эвристикой. При решении такого рода задач описание мнения субъекта, как правило, превышает по объему имеющееся в распоряжении исследователя описание самого субъекта. В этом случае разумным представляется использование обратного ДСМ-рассуждения, анализирующего сходство мнений субъектов и на основании этого анализа выявляющего сходство самих субъектов, имеющих общие мнения (это отношение представлено $*_3\Leftarrow$). Существенной при этом оказывается возможность структурированного представления мнений (подобно тому как было структурировано описание индивидуумов на основании постулата поведения).

Использующийся вариант семантики ДСМ-метода для анализа и прогнозирования мнений[17] опирается на представление об опросе как множестве ответов на вопросы по соответствующей теме T. В этом случае тема T характеризуется утверждениями $p_1, ..., p_n$ – корнями вопросов (параметрами опроса) из множества $P = \{p_1, ..., p_n\}$, называемого каркасом темы. В результате опроса отдельно устанавливается отношение респондентов к элементам каркаса и к теме в целом. Задана функция оценки $v^{(i)}[p_j] = \nu_j^{(i)}$ с областью значений $V_m = \{0, 1/m-1,..., m-1/m-2, 1\}$ в общем случае m-значного опроса $(m\geq 2)$[18], $i = 1, ..., m^n, j = 1, ..., n$. Каждому элементу p_j ($j = 1, ..., n$) каркаса P соответствует вопрос $?p_j$ «Какова оценка ν корня вопроса p_j ?», $\nu \in V_m$, ответом же является высказывание $J_\nu p_j$. $J_\nu p_j = t$, если $v[p_j] = \nu$; $J_\nu p_j = f$, если $v[p_j] \neq \nu$.

Тогда ответом *i*-го респондента по теме Т будем называть конъюнкцию $\varphi_i = J_{\nu_1^{(i)}} p_1 \& \ldots \& J_{\nu_n^{(i)}} p_n$, где φ_i – метасимвол, V – предикат графического равенства формул. Такой ответ представляет собой понимание *i*-м респондентом темы Т. Множество членов этой конъюнкции обозначим $[\varphi_i]=\{ J_{\nu_1^{(i)}} p_1, \ldots, J_{\nu_n^{(i)}} p_n\}$ и будем называть составом мнения.

Здесь необходимо подчеркнуть, что для формализации *m*-значного закрытого опроса задаются *m*-значная логика J_m и исчисление эквивалентных формул ИЭФ-J_m[19]. Областью значений переменных в J_m-логиках является V_m, помимо бинарных связок $\&'$, $\vee'$, $\supset'$ с областью значений {0, 1} (на множестве значений {0, 1} эти связки совпадают с двузначными связками &, $\vee$, $\supset$), задаются также унарные связки J_ν ($\nu \in V_m$), которые для пропозициональных переменных p определяются следующим образом: $J_\nu p = 1$, если $v[p] = \nu$; $J_\nu p = 0$, если $v[p] \neq \nu$. Функция оценки $v[\varphi]$ формул логики J_m определяется индуктивно по сложности формулы φ и принимает значение из {0, 1}. Таким образом, для формул φ логики J_m имеет место $v^{(i)}[\varphi]=\varphi(v^{(i)}[p_1], \ldots, v^{(i)}[p_n])$, где $i=1, \ldots, m^n$, а $p_1, \ldots, p_n$ – все переменные, входящие в φ. Система этих равенств определяет функцию F_φ такую, что она отображает множество V_m в {0, 1}, т. е. $F_\varphi: V_m \to \{0, 1\}$. Для представления эквивалентных формул (реализующих одну и ту же функцию) строится ИЭФ-J_m, которое является модификацией ИЭФ двузначной логики. Определенные выше мнения респондентов представляют собой J_m-максимальные конъюнкции $C_i = J_{\nu_1^{(i)}} p_1 \& \ldots \& J_{\nu_n^{(i)}} p_n$ J_m-атомов $J_{\nu_k^{(i)}} p_k$ (где $k=1, \ldots, n$, $\nu_k^{(i)} \in V_m$, а $i=1, \ldots, m^n$): 1) для каждой p_k в C_i входит $J_{\nu_k^{(i)}} p_k$, причем без повторений, $k=1, \ldots, n$; 2) если $\nu_k^{(i)} \neq \nu_j^{(i)}$, $J_{\nu_k^{(i)}} p_k$ и $J_{\nu_j^{(i)}} p_k$ не входят в C_i одновременно. J_m-элементарной конъюнкцией логики J_m называется конъюнкция J_m-атомов. Ниже мы рассмотрим процедуры преобразования мнений, представляющих собой максимальные конъюнкции логики высказываний J_m, для реализации эвристики «булева алгебра – аналогия – абдукция», *AlAnAb*.

Множество всех возможных ответов по теме Т с каркасом *P* обозначим *K*. При этом число элементов этого множества $|K|=m^n$, поскольку каждой J_m-максимальной конъюнкции взаимно однозначно соответствует *m*-значный (*n*-мерный) вектор $\vec{\sigma}^{(i)} =$

$= \langle \sigma_1^{(i)}, ..., \sigma_n^{(i)} \rangle$, где $v^{(i)}[p_j]=\sigma_j^{(i)}$, $i=1, \ldots, m^n$, $j=1, \ldots, n$. Заметим, что число респондентов может превышать m^n, поскольку различные респонденты могут иметь одинаковые ответы, при этом число различных ответов может быть меньше m^n.

$$K = \{\varphi_i \mid \varphi_i \rightleftharpoons J_{\nu_1^{(i)}}\, p_1 \& ... \& J_{\nu_n^{(i)}}\, p_n, v^{(i)}[p_j] = \nu_j^{(i)}, \nu_j^{(i)} \in V_m, j = 1, ..., n, i = 1, ..., m^n\}.$$

Итак, анализ мнений средствами ДСМ-метода АПГ осуществляется в соответствии со следующей стратегией. Формулируется тема мнения, задается система вопросов, раскрывающих содержание темы – каркас темы. Оценка эмпирического отношения $C \Rightarrow_1 Q$ (субъект – мнение) есть оценка отношения к теме в целом, Q – состав мнения субъекта C (множество $\{ J_{\nu_1^{(i)}}\, p_1, ..., J_{\nu_n^{(i)}}\, p_n\}$ образующих (атомов) мнения с оценками, $Q = [\varphi]$).

Напомним, что в «классическом» варианте ДСМ-метода для оценки фактов используется 4 типа истинностных значений: 1 – фактическая истина, –1 – фактическая ложь, 0 – фактическое противоречие» и τ – неопределенность. Пусть даны конечные множества $\mathbf{U}^{(1)} = \{d_1, ..., d_r\}$ – множество дифференциальных признаков индивидуумов (в соответствии с постулатом поведения), множество возможных ответов на вопросы каркаса $\mathbf{U}^{(2)} = \{\psi \mid (\psi \rightleftharpoons J_{\nu_i}\, p_i) \& (\nu_i \in \{1, -1, 0, \tau\}), i = 1, ..., n\}$, $|\mathbf{U}^{(2)}| = 4n$ (для указанных четырех типов истинностных значений).

Массив начальных данных (БФ) содержит утверждения типа «высказывание “субъект C_i имеет мнение $[\varphi_i]$” имеет истинностное значение $\langle \nu, 0 \rangle$ в его отношении к теме опроса» ($J_{\langle \nu, 0 \rangle}(C_i \Rightarrow_1 [\varphi_i])$ в ДСМ-языке[20]), $\nu \in \{1, -1, 0, \tau\}$. В результате применения правил индуктивного вывода (п.п.в.-1 для обратного метода) порождаются гипотезы вида $J_{\langle \nu, n \rangle}([\psi_i]\ {}_3\!\Leftarrow C'_i)$, n – номер шага вычислений, выражающий степень правдоподобия истинностного значения, $n>0$. Это выражение означает, что «высказывание “мнение ψ_i есть следствие характеристик субъекта C'_i” имеет истинностное значение $\langle \nu, n \rangle$». Как и выше, $J_{\langle \nu, n \rangle}\varphi = t$, если $v[\varphi] = \langle \nu, n \rangle$; $J_{\langle \nu, n \rangle}\varphi = f$, если $v[\varphi] \neq \langle \nu, n \rangle$, $v[\varphi]$ есть функция оценки, $\langle \nu, n \rangle$ представляет «внутренние» истинностные значения фактов и гипотез, t, f – «внешние» истинностные значения двузначной логики. Порожденные детерминанты мнений в дальнейшем используются для прогнозирования мнений с помощью правил вывода по аналогии (п.п.в.-2), а также могут служить основанием для построения модели структуры изу-

чаемого социума. Здесь C_i, C'_i, $[\varphi_i]$, $[\psi_i]$ – константы, C_i, $C'_i \in 2^{U^{(1)}}$, $[\varphi_i]$, $[\psi_i] \in 2^{U^{(2)}}$, высказывания $J_{\langle v,0\rangle}(C \Rightarrow_1 Q)$ суть факты, $J_{\langle v,n\rangle}(C \Rightarrow_j Q)$ ($j = 1, 2$, $n>0$) – гипотезы.

Опишем коротко, как осуществляется индуктивный анализ сходства мнений, а также вывод по аналогии и абдуктивное принятие гипотез в ДСМ-методе; детальное описание можно найти в литературе[21].

(1) Предикат простого обратного положительного сходства $\tilde{\breve{M}}^{+}_{a,n}(V,W,k)$, использующийся при формулировке п.п.в.-1, распознает локальное сходство $(\bigcap_{i=1}^{k} C_i=C')\&(C'\neq\varnothing)\&((\bigcap_{i=1}^{k} Q_i=Q')\&(Q'\neq\varnothing)$ на множестве (+)-примеров $J_{(1,n)}(C_i \Rightarrow_1 Q_i)$, $i = 1,..., k$, ($k\geq 2$), которое является основанием для правдоподобного вывода.

(2) Одной из важнейших является подформула предиката, выражающая условие исчерпываемости – требование рассмотрения всех имеющихся в БФ подходящих мнений.

(3) Предикат описывает эмпирическую зависимость (ЭЗ) $\forall X\forall Y((J_{(1,n)}(X\Rightarrow_1 Y)\&\forall U(J_{(1,n)}(X\Rightarrow_1 U)\rightarrow U\subseteq Y)\&W\subseteq Y)\rightarrow(V\subset X\&V\neq\varnothing))$ типа «сходство мнений субъектов в (+)-примерах влечет сходство самих субъектов и притом для всех рассматриваемых мнений (условие исчерпываемости мнений в (+)-примерах)».

Непараметрический предикат простого обратного положительного сходства $\breve{M}^{+}_{a,n}(V,W)$ выполняется, если существуют k (+)-примеров таких, что их сходство выразимо посредством $\tilde{\breve{M}}^{+}_{a,n}(V,W,k)$. Предикат простого обратного отрицательного сходства $\breve{M}^{-}_{a,n}(V,W)$ формулируется симметрично. Оба предиката являются взаимнофальсифицирующими: на (n+1)-м шаге ДСМ-рассуждений порождается гипотеза $J_{\langle 1,\ n+1\rangle}(W_3\Leftarrow V)$, если имеет место $\breve{M}^{+}_{a,n}(V,W)\&\neg\breve{M}^{-}_{a,n}(V,W)$, и наоборот. Если выполнены одновременно $\breve{M}^{+}_{a,n}(V,W)$ и $\breve{M}^{-}_{a,n}(V,W)$, порождается противоречивая гипотеза $J_{\langle 0,\ n+1\rangle}(W_3\Leftarrow V)$.

Случаи неопределенности в БФ $J_{(\tau,n)}(C\Rightarrow_1 Q)$ уточняются с помощью п.п.в.-2 – выводов по аналогии, использующих гипотезы о причинах (результаты применения п.п.в.-1). Если C покрывается

множеством C_i', i =1, ..., k, таких, что $C_i' \subset C$, C_i' есть (+)-причина Q_i (на n-ом шаге) и $\bigcup_{i=1}^{k} Q_i = Q$, но (–)-причины (для Q_i) не содержатся в C, то порождается (+)-гипотеза: «высказывание "объект C есть причина наличия множества свойств Q" имеет истинностное значение $\langle 1, n+1\rangle$», $J_{\langle 1, n+1\rangle}(C \Rightarrow_1 Q)$. Аналогично порождаются гипотезы для типов истинностных значений –1, 0, τ.

Завершающим этапом ДСМ-рассуждения является абдуктивное объяснение начального состояния БФ, т. е. принятие порожденных гипотез первого и второго рода на основании проверки так называемых аксиом каузальной полноты предметной области (социума). Смысл АКП$^{(+)}$ состоит в следующем: для каждого (+)-факта «объект C обладает множеством свойств Q» из начального состояния БФ, $J_{\langle 1, 0\rangle}(C \Rightarrow_1 Q)$ существуют (+)-причины C′ для фрагментов Q′, $J_{\langle 1, n\rangle}(Q'_{\ 3} \Leftarrow C')$ такие, что $C' \subset C$ и свойства Q полностью покрываются фрагментами Q′, $\cup Q' = Q$. Аналогично формулируется АКП$^{(-)}$.

Рассматриваемая реализация эвристики «сходство – аналогия – абдукция» для задачи анализа мнений опирается на предложенное ранее[22] определение m-значного закрытого опроса ($m \geq 2$) O_m по теме T средствами дедуктивной m-значной логики J_m как $O_m = \langle J_m, P, \Sigma, K', R\rangle$. Формулы логики J_m из непротиворечивого множества $\Sigma = \{\psi_1, ..., \psi_s\}$ выражают логические зависимости между элементами каркаса $P = \{p_1, ..., p_n\}$ и используются при вычислении степени непротиворечивости опроса, частично характеризующей рациональность мнений[23]. Множество респондентов $R = \{X \mid \exists \varphi J_{\langle \nu, 0\rangle}(X \Rightarrow_1 [\varphi]) \& \varphi \in K'\}$ ($[\varphi] = \{J_{\nu_1} p_1, ..., J_{\nu_n} p_n\}$; $\nu, \nu_1, ..., \nu_n \in V_m$) соответствует множеству стабилизированных ответов $K' \subseteq K$, которое не изменяется при расширении множества опрашиваемых. Таким образом, достижение K' есть конструктивное порождение насыщенной выборки – одной из основных идей обоснованной теории (см. примеч. 4).

Если вспомнить, что оценки относительно элементов каркаса P и темы T формируются независимо, опрос для каркаса («внутренний») может быть m-значным (соответственно, используется логика J_m), опрос по теме («внешний») может быть при этом l-значным (логика J_l). Опрос определяется расширенно как $O_{m,l} = \langle J_m, J_l, P, \Sigma, K', R\rangle$. В этом случае в определении R $\nu \in V_m$, ν_1, ..., $\nu_n \in V_l$. Заметим, что в качестве логического аппарата таких опросов мо-

жет использоваться аргументационная семантика[24], однако это возможно не для любых m и l. Более того, и в общем случае для m и l должна быть предложена осмысленная социологическая интерпретация.

Рассмотрим различные варианты «внутренних» и «внешних» опросов и сравним результаты, полученные в результате применения различных реализаций – алгебраической и ДСМ-метода – эвристической схемы «сходство – аналогия – абдукция». Отметим, что для рассматриваемого варианта анализа мнений имеет место выполнение следующего условия (которое может быть названо онтологической аксиомой и представляет собой определение мнения респондента X): $\forall X \exists \nu_1 ... \exists \nu_n (J_{\langle \nu, 0 \rangle}(X \Rightarrow_1 \{ J_{\nu_1} p_1, \ ..., \ J_{\nu_n} p_n\}) \rightarrow$ $(J_{\nu_1} p_1 \& ... \& J_{\nu_n} p_n))$, где $\nu \in V_m$, $\nu_1, ..., \nu_n \in V_l$.

1. Простейший вариант: внутренние оценки – булевские (m=2), опрос по теме – булевский (l=2).

В этом опросе для каждого элемента каркаса (так же как и для отношения к теме) респонденту предлагается выбрать один из ответов: «да» – 1 или «нет» – 0. Тогда мнению φ_i респондента X_i относительно каркаса $P = \{p_1, ..., p_n\}$ соответствует булевский вектор $\vec{\sigma}^{(i)} = \langle \sigma_1^{(i)}, ..., \sigma_n^{(i)} \rangle$, где $\sigma_j^{(i)} = 0, 1, j = 1, ..., n$, а $i = 1, ..., k$ (число различных мнений, которое не больше числа респондентов, $k \le 2^n$). Если $\vec{\sigma}^{(i)}$ соответствует атомарной оценке $v^{(i)}$, то $\vec{\sigma}^{(i)} = \langle v^{(i)}[p_1], ..., v^{(i)}[p_n] \rangle$. Пусть оценка отношения к теме для i-го респондента равна $\sigma^{(i)}$, $\sigma^{(i)} = 0, 1$, тогда каждому мнению φ_i (вектору $\vec{\sigma}^{(i)}$) отвечает $\sigma^{(i)}$. Таким образом, задается булевская функция, характеризующая связь между ответами на вопросы, составляющие каркас темы, и отношением к теме в целом.

Обозначим $\Phi_0^{(1)}$ множество всех таких мнений φ_i, что $\sigma^{(i)} = 1$, т. е. отношение к теме T в начальном состоянии БФ положительное, $\Phi_0^{(1)} = \{\varphi \mid \exists X J_{\langle 1, 0 \rangle}(X \Rightarrow_1 [\varphi]) \& \ (\varphi \in K')\}$, $\Phi_0^{(1)} = \{\varphi_1, ..., \varphi_s\}$. Для удобства представим φ_i ($i = 1, ..., s$) в виде $p_1^{\sigma_1^{(i)}} \ \& \ ... \ \& \ p_n^{\sigma_n^{(i)}}$ в соответствии с принятой в булевой алгебре нотацией $p^{\sigma}=p$, $\sigma=1$, $p^{\sigma}=\neg p$, $\sigma=0$. Соответственно, множество субъектов, имеющих мнения из $\Phi_0^{(1)}$, обозначим $R_0^{(1)} = \{X_1, ..., X_m\}$, $R_0^{(1)} = \{X \mid J_{\langle 1,0 \rangle}(X \Rightarrow_1 [\varphi_i]) \& (\varphi_i \in \Phi_0^{(1)})\}$ ($i = 1, ..., s$).

Стратегия дальнейшего исследования отношения к теме (*AlAnAb*) восходит к стратегии, используемой в QCA (см. примеч. 7). Выражение $\varphi_1 \vee ... \vee \varphi_s$ представляет собой совершенную дизъюнк-

тивную нормальную форму (далее – СДНФ) для положительного отношения к теме (напомним, что мнения φ – максимальные конъюнкции соответствующей логики J_m, в данном случае – двузначной). Преобразуем стандартным образом (с помощью алгоритма Куайна) СДНФ к сокращенной ДНФ[25] $\partial(\varphi_1 \vee \ldots \vee \varphi_s) \quad \chi_1 \vee \ldots \vee \chi_r$. Рассмотрим множество импликант $[\partial\varphi] = \{\chi_1, \ldots, \chi_r\}$. Каждой импликанте χ_j поставим в соответствие такое множество $\Phi_0^{(1)}{}_j$ мнений φ, что φ покрывается импликантой χ_j, $\Phi_0^{(1)}{}_j = \{\varphi \mid \chi_j \quad \varphi\}$, $j = 1, \ldots, r$. Соответственно, множество субъектов, мнение которых есть элемент $\Phi_0^{(1)}{}_j$, обозначим $R_0^{(1)}{}_j = \{X \mid J_{\langle 1,0\rangle}(X \Rightarrow_1 [\varphi_q]) \& \varphi_q \in \Phi_0^{(1)}{}_j\}$, $R_0^{(1)}{}_j = \{X_{j_1}, \ldots, X_{j_h}\}$. Сходство элементов $R_0^{(1)}{}_j$ – всех X таких, что их мнение покрывается импликантой χ_j, – обозначим V_j', $V_j' = \bigcap_{k=1}^{h} X_{j_k}$.

Для реализации приведенной выше схемы будем считать, что отношение каузальности $C(V_j', \chi_j)$ (аналог $_3\Leftarrow$ в обратном ДСМ-методе) представлено парами $\langle V_j', \chi_j\rangle$ $(V_j' \neq \varnothing, j = 1, \ldots, r)$, т. е. мнение $\varphi_q \in \Phi_0^{(1)}{}_j$ $(J_{\langle 1,0\rangle}(X \Rightarrow_1 [\varphi_q]))$ объясняется наличием множества характеристик $V_j' \subseteq X$, V_j' – детерминанта мнения φ_q.

Подчеркнем несколько обстоятельств. В рассмотренном варианте мы исходим из непротиворечивого представления мнений в исходной БФ: мы полагаем, что респонденты с позитивным (1) и негативным (0) отношением к теме имеют разные мнения, $\Phi_0^{(1)} \cap \Phi_0^{(0)} = \varnothing$, где $\Phi_0^{(1)}$ определено выше, а $\Phi_0^{(0)} = \{\varphi \mid \exists X J_{\langle 0,\ 0\rangle}(X \Rightarrow_1 [\varphi]) \& (\varphi \in K')\}$. Мы могли бы также породить соответствующие импликанты и отношения причинности для негативного отношения к теме. Но поскольку в булевском опросе существует лишь два типа истинностных значений (как внешних, так и внутренних), противоречия, могущие возникнуть в случае совпадения некоторых мнений у респондентов с противоположным отношением к теме, не могут быть описаны в рамках предложенного формализма.

Далее описанная процедура представляет собой лишь одну из частей общей эвристической схемы, а именно поиск сходства средствами булевой алгебры. Рассмотрим, как соотносятся гипотезы о причинах, полученные в результате применения описанной схемы и традиционного поиска сходства в ДСМ-методе. Пусть в результате применения индуктивных правил п.п.в.-1 обратного ДСМ-метода получено множество гипотез $J_{\langle 1,1\rangle}([\psi_h]\ {}_3\Leftarrow V_h)$, $h = 1, \ldots, t$. Вследствие того что опрос булевский, мы имеем дело с несиммет-

ричным вариантом ДСМ-метода: гипотезы для негативного восприятия темы не порождаются и, соответственно, (+)-гипотезы не фальсифицируются. Будем считать, что полученные ψ_1, ..., ψ_t образуют сокращенную ДНФ для описания положительного отношения к теме $\partial\psi = \psi_1 \vee \ldots \vee \psi_t$, а множество импликант есть $[\partial\psi] = \{\psi_1, \ldots, \psi_t\}$. Охарактеризуем возможные соотношения множеств $[\partial\psi]$ и $[\partial\varphi] = \{\chi_1, \ldots, \chi_r\}$.

Рассмотрим для начала в каждом случае сходство мнений (без выявления сходства самих субъектов, выражающих мнение). Заметим сразу, что на каждом шаге преобразования СДНФ в сокращенную ДНФ (выполнения алгоритма Куайна) для мнений выполняется условие исчерпываемости.

А. Пусть $\Phi_0^{(1)} = \{\varphi_1, \varphi_2\}$, φ_1, $p_1 \& p_2$ (в дальнейшем для простоты знак & в булевских выражениях мы будем опускать, записывая p_1p_2), $\varphi_2 = p_1\neg p_2$. Сокращенная ДНФ $\partial(\varphi_1 \vee \varphi_2) = p_1$, и множество импликант, порожденных средствами булевой алгебры, $[\partial\varphi] = \{p_1\}$. Но и множество импликант, составляющих фрагменты ДСМ-гипотез, есть $[\partial\psi] = \{p_1\}$. Рассмотрим теперь отношение причинности. Сходство субъектов, мнения которых покрываются импликантой p_1, в рассматриваемом случае есть сходство всех респондентов $R_0^{(1)} = \{X_1, \ldots, X_k\}$, $V' = \bigcap_{i=1}^{k} X_i$. Если $V' \neq \varnothing$, отношение каузальности C представлено парой $\langle V', p_1\rangle$. Но в этом случае порождается и ДСМ-гипотеза $J_{\langle 1,1\rangle}(\{p_1\} \Leftarrow_3 V')$. Однако ДСМ-рассуждение может породить и другие гипотезы: если хотя бы одно мнение (φ_1 или φ_2) имеет больше двух респондентов, то порождаются также гипотезы $J_{\langle 1,1\rangle}(\{p_1, p_2\} \Leftarrow_3 V_1')$ и/или $J_{\langle 1,1\rangle}(\{p_1, \neg p_2\} \Leftarrow_3 V_2')$, где V_1' – сходство респондентов, имеющих мнение φ_1, V_2' – сходство респондентов с мнением φ_2.

Б. Пусть $\Phi_0^{(1)} = \{\varphi_1, \varphi_2\}$, $\varphi_1 = p_1p_2p_3$, $\varphi_2 = p_1\neg p_2\neg p_3$. Тогда СДНФ совпадает с сокращенной ДНФ $\partial(\varphi_1 \vee \varphi_2) = p_1p_2p_3 \vee p_1\neg p_2\neg p_3$, и множество импликант, порожденных средствами булевой алгебры, $[\partial\varphi] = \{p_1p_2p_3,\ p_1\neg p_2\neg p_3\}$. Соответственно, причинность задается двумя парами $\langle V_1', p_1p_2p_3\rangle$, $\langle V_2', p_1\neg p_2\neg p_3\rangle$, где V_1' – сходство респондентов с мнением φ_1, V_2' – сходство респондентов с мнением φ_2. ДСМ-метод, помимо гипотез, совпадающих с этими двумя отношениями (если для каждого мнения больше одного респондента), может породить также гипотезу $J_{\langle 1,1\rangle}(\{p_1\} \Leftarrow_3 V')$, где $V' \neq \varnothing$, $V' = V_1' \cap V_2'$, т. е. V' есть сходство всех респондентов с положительным отношением к теме.

В. Пусть $\Phi_0^{(1)} = \{\varphi_1, \varphi_2\}$, $\varphi_1 \rightleftharpoons p_1p_2$, $\varphi_2 \rightleftharpoons \neg p_1\neg p_2$. Тогда, как и в предыдущем случае, СДНФ совпадает с сокращенной ДНФ $\partial(\varphi_1 \vee \varphi_2) \rightleftharpoons p_1p_2 \vee \neg p_1\neg p_2$, и множество импликант, порожденных средствами булевой алгебры $[\partial\varphi] = \{p_1p_2, \neg p_1\neg p_2\}$. Если респондентов для каждого мнения не меньше двух, алгебраические гипотезы и ДСМ-гипотезы совпадают.

Г. Мы видели, что в рассмотренных выше случаях множество ДСМ-гипотез либо совпадает с множеством алгебраических гипотез, либо включает также дополнительные гипотезы. Однако в самом общем случае ДСМ-гипотез оказывается больше, чем алгебраических, поскольку ДСМ-гипотезы могут порождаться на промежуточных шагах преобразования СДНФ в сокращенную. Соответственно, обратная ситуация, когда алгебраических гипотез больше, чем ДСМ-гипотез, по-видимому, невозможна.

Пусть $\Phi_0^{(1)} = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$, $\varphi_1 \rightleftharpoons p_1p_2p_3p_4$, $\varphi_2 \rightleftharpoons p_1p_2p_3\neg p_4$, $\varphi_3 \rightleftharpoons p_1\neg p_2\neg p_3p_4$, $\varphi_4 \rightleftharpoons p_1\neg p_2\neg p_3\neg p_4$. Тогда сокращенная ДНФ $\partial(\varphi_1 \vee \varphi_2 \vee \varphi_3 \vee \varphi_4) \rightleftharpoons p_1p_2p_3 \vee p_1\neg p_2\neg p_3$, и множество импликант, порожденных средствами булевой алгебры, $[\partial\varphi] = \{p_1p_2p_3, p_1\neg p_2\neg p_3\}$. Однако множество ДСМ-импликант, представляющих ДСМ-гипотезы, есть $[\partial\psi] = \{p_1, p_1p_2p_3, p_1\neg p_2\neg p_3, p_1p_4, p_1\neg p_4\}$. Если соответствующие сходства респондентов не пусты, мы имеем две алгебраические гипотезы и пять ДСМ-гипотез.

Разумеется, существенный интерес представляет сравнение результатов, полученных на основании эмпирических данных.

2. Рассмотренный вариант легко переносится на случай опроса, когда внутренние оценки – m-значные (m=≥3), опрос по теме – булевский (l=2).

Как уже говорилось, средством формализации m-значного опроса являются J_m-логики (см. примеч. 19). Для этих логик доказывается теорема о представимости всякой не эквивалентной 0 формулы φ логики высказываний J_m посредством совершенной дизъюнктивной нормальной формы J_m-СДНФ (J_m-С$\partial\varphi$) и притом единственным образом, а именно: $\varphi(p_1, ..., p_n) \leftrightarrow \bigvee'_{\varphi(\vec{\sigma}^{(i)})=1} (J_{\sigma_1^{(i)}} p_1 \&' ... \&' J_{\sigma_n^{(i)}} p_n)$, где $(\psi \leftrightarrow \chi)$ $((\psi \supset' \chi) \&' (\chi \supset' \psi))$, $\vec{\sigma}^{(i)} = \langle \sigma_1^{(i)}, \ldots, \sigma_n^{(i)} \rangle$, а $\sigma_j^{(i)} \in V_m$, j=1, ..., n. Далее формулируется обобщение алгоритма Куайна для перевода J_m-СДНФ в сокращенные J_m-ДНФ. «Обобщенное склеивание» и поглощение в этом случае выглядят следующим образом (C_1', ..., C_m', C', C'' – J_m-элементарные конъюнкции):

(a) $(J_0\mathrm{p}\&'\mathrm{C}_1')\vee'(J_{\frac{1}{m-1}}\mathrm{p}\&'\mathrm{C}_2')\vee'\ldots\vee'(J_{\frac{m-2}{m-1}}\mathrm{p}\&'\mathrm{C}'_{m-1})\vee'(J_1\mathrm{p}\&'\mathrm{C}'_m)\leftrightarrow$
$\leftrightarrow(J_0\mathrm{p}\&'\mathrm{C}_1')\vee'\ldots\vee'(J_1\mathrm{p}\&'\mathrm{C}'_m)\vee'(\mathrm{C}_1'\&'\ldots\&'\mathrm{C}'_m)$,

(b) $(\mathrm{C}_1'\&'\mathrm{C}'')\vee'\mathrm{C}''\leftrightarrow\mathrm{C}''$.

Применяя последовательно, пока это возможно, преобразования (a) и (b) к формуле φ логики J_m (в нашем случае – к дизъюнкции мнений $\varphi_i \rightleftharpoons J_{\nu_1^{(i)}}\mathrm{p}_1\&\ldots\&J_{\nu_n^{(i)}}\mathrm{p}_n$, $\mathrm{v}^{(i)}[\mathrm{p}_j]=\nu_j^{(i)}$, $\nu_j^{(i)}\in\mathrm{V}_m$, $j=1,\ldots,n$, $i=1,\ldots,m^n$ из начального состояния БФ, $\Phi_0^{(1)}$), получим ∂φ – сокращенную ДНФ формулы φ. Полученные импликанты используются для определения отношения причинности аналогично описанной выше схеме. Таким образом, соотношение между ДСМ-гипотезами и алгебраическими гипотезами, использующими импликанты, сохраняется.

Полная эвристическая процедура предусматривает после получения гипотез о причинах использование вывода по аналогии для предсказания неизвестных свойств у объектов (в нашем случае – мнений респондентов), а также абдуктивное объяснение имеющихся фактов. Использование булевой алгебры для поиска сходства этих фактов сообщает последующим шагам несколько особенностей. Главная особенность состоит в предположении о «замкнутости мира», поскольку дизъюнктивные нормальные формы строятся на основании истинностной таблицы. Стало быть, полностью корректным построение причинных отношений может быть лишь в случае $K' = K$. Далее имеющиеся примеры положительного отношения к теме всегда объясняются – по самой процедуре построения причинных гипотез, – если только сходство примеров, мнения которых покрываются одной импликантой, не пусто. Предикат объяснения примеров из БФ – $J_{\langle 1,0\rangle}(\mathrm{X}\Rightarrow_1\mathrm{Y})$ – импликантами может быть записан в виде $E(\mathrm{X},\mathrm{Y}) \rightleftharpoons$ $\rightleftharpoons\exists\mathrm{V}'\exists\chi((\mathrm{V}'\subseteq\mathrm{X})\&(\mathrm{V}'\neq\varnothing)\&([\chi]\subseteq\mathrm{Y})\&C(\mathrm{V}',\chi)\&J_{\langle 1,0\rangle}(\mathrm{X}\Rightarrow_1\mathrm{Y}))$, где $[\chi]$ – множество атомов, входящих в импликанту χ, $[\chi]=\{J_{\nu_{i_1}}p_{i_1},\ldots,J_{\nu_{i_k}}p_{i_k}\}$. Но каждый пример положительного отношения к теме $J_{\langle 1,0\rangle}(\mathrm{X}\Rightarrow_1\mathrm{Y})$ (с оценкой 1 в истинностной таблице) был использован при порождении импликант, так что для мнения Y каждого респондента X найдется импликанта χ, его покрывающая ($[\chi]\subseteq$Y). Соответственно, если респонденты, мнение которых покрывается этой импликантой, сходны между собой ($\mathrm{V}'\neq\varnothing$), фрагмент V′ опи-

сания респондента X, являющийся частью описанного выше отношения причинности $C(V', \chi)$, также обязательно входит в описание респондента $(V' \subseteq X)$.

Коль скоро речь идет о полном описании всех возможных мнений, потребность в использовании вывода по аналогии может возникнуть лишь для расширения имеющейся БФ – когда возникнет потребность определить отношение к теме для новых респондентов, высказавших свое мнение относительно элементов каркаса *P*. Поскольку в рассматриваемой схеме порождаются импликанты только для положительных примеров (отношения к теме), предикат для вывода по аналогии может учитывать лишь вхождение соответствующих фрагментов описания субъекта V и покрытие мнения импликантами χ: $\Pi^{+}(X, Y) \rightleftharpoons \exists V' \exists \chi((V' \subseteq X) \& (V' \neq \varnothing) \& ([\chi] \subseteq Y) \& C(V', \chi))$. Соответствующее правило для вывода по аналогии формулируется аналогично п.п.в.-2 ДСМ-метода АПГ, описанным выше, в результате порождается гипотеза $J_{\langle 1,2 \rangle}(X \Rightarrow_1 Y)$. Добавление полученных доопределенных примеров к БФ не может изменить порожденных отношений причинности, так как и импликанты, и сходство описания соответствующих этим импликантам респондентов остаются прежними (по предположению мы имеем полное представление возможных мнений в БФ, а доопределение происходит по включению соответствующих фрагментов в описание субъекта, $V' \subseteq X \& V' \neq \varnothing$). Следовательно, отпадает необходимость в циклическом повторении процедур «сходство – аналогия», что коренным образом отличает алгебраический подход от подхода ДСМ-метода.

Рассмотрим случай, когда $K' \subset K$, т. е. в БФ нет примеров для отношения к теме при наличии мнения из $K \backslash K'$ и, соответственно, нет примеров описания респондентов с такими мнениями. Сокращенная ДНФ строится на основании мнений из $\Phi_0^{(1)}$, т. е. все примеры мнений из $K \backslash \Phi_0^{(1)}$, в том числе и отсутствующие мнения из $K \backslash K'$, оказываются нулями соответствующей булевской функции (в случае рассмотренного выше полного описания такими нулями были только мнения из $\Phi_0^{(0)}$). Следовательно, мнения из $K \backslash K'$ не будут покрываться полученными импликантами, и доопределить новые примеры респондентов с такими мнениями корректно принципиально невозможно.

В свете сказанного можно предложить следующую эвристическую стратегию предсказания отношения к теме для новых рес-

пондентов. Прежде всего следует проверить, встречалось ли соответствующее мнение в исходной БФ. Для этого можно построить сокращенную ДНФ, описывающую условия наличия примеров мнений (или, симметрично, их отсутствия). СДНФ в этом случае включает все мнения из $K' = \Phi_0^{(1)} \cup \Phi_0^{(0)}$ (1 для булевской функции наличия примеров в БФ). Для нового мнения проверяем, покрывается ли оно полученными импликантами (разумеется, это можно сделать и простым сравнением нового мнения с элементами K'). Если мнение Y уже встречалось, проверяем для респондента X выполнимость предиката $\Pi^+(X, Y)$. Если предикат выполняется, порождаем гипотезу $J_{\langle 1,2\rangle}(X \Rightarrow_1 Y)$. Если нет – гипотезу $J_{\langle 0,2\rangle}(X \Rightarrow_1 Y)$, поскольку для замкнутой таблицы все, что не является 1, является 0.

Если же мнение $Y = [\varphi]$, $\varphi \in K \setminus K'$, можно проверить вхождение в X каких-либо детерминант V_j' ($V_j' \neq \varnothing$, $j = 1, ..., r$) из множества $\{V_1', ..., V_r'\}$, соответствующего множеству импликант $[\partial\varphi] = \{\chi_1, ..., \chi_l\}$. Хотя эти детерминанты являются детерминантами мнений, но мнений из $\Phi_0^{(1)}$ с положительным отношением к теме. Возможно, они отвечают одновременно и за отношение к теме (хотя это недостаточно обоснованно), и в этом случае можно условно принять гипотезу $J_{\langle 1,2\rangle}(X \Rightarrow_1 Y)$. Можно слегка усилить эту гипотезу, если одновременно проверить вхождение в описание X (–)-детерминант, порожденных для мнений из $\Phi_0^{(0)}$ (необходимости в таких детерминантах в предыдущих случаях не было ввиду замкнутости мира). Отсутствие вхождения (–)-детерминант подкрепляет приведенную гипотезу. Еще более сильным вариантом было бы наличие сходства для всех респондентов с положительным отношением к теме и вхождение этого сходства в рассматриваемый пример. Приведенные соображения являются не более чем предложением для эмпирического исследователя.

Все сказанное свидетельствует о серьезных различиях в результатах применения эвристической схемы «сходство – аналогия – абдукция» в алгебраическом и ДСМ-подходах: меньшее число гипотез о причинах в первом случае, большее число объясненных исходных примеров, невозможность формального доопределения примеров с ранее не встречавшимися мнениями. Однако за пределами булевского опроса некоторые расхождения могут быть дополнительно сглажены.

3. Рассмотрим опрос для стандартной 4-значной ДСМ-логики, когда отношение к теме характеризуется оценками $\nu \in \{+1, -1, 0, \tau\}$ (см. выше). Опрос относительно элементов каркаса может быть m-значным с соответствующим использованием J_m-логик для порож-

дения импликант и отношения каузальности. Сходство мнений в ДСМ-методе при этом есть теоретико-множественное сходство составов мнений и не зависит от значности опроса. Без ограничения общности можно считать, что и внутренний опрос представлен вариантами ответов +1 («да»), –1 («нет»), 0 («и да, и нет»), τ (не определено). Ответом *i*-го респондента по теме опроса T будет максимальная конъюнкция $\varphi_i = J_{\nu_1^{(i)}} p_1 \& \ldots \& J_{\nu_n^{(i)}} p_n$, где $\nu_i^{(j)} \in \{\pm 1, 0, \tau\}$, $i = 1, \ldots, n; j = 1, \ldots, 4^n$.

Пусть в БФ представлено множество респондентов $R_0 = R_0^{(+1)} \cup R_0^{(-1)} \cup R_0^{(0)} \cup R_0^{(\tau)}$, где $R_0^{(\nu)} = \{X | \exists \varphi J_{\langle \nu, 0 \rangle}(X \Rightarrow_1 [\varphi]) \& (\varphi \in K')\}$, $\nu \in \{+1, -1, 0\}$, $R_0^{(\tau)} = \{X | \exists \varphi J_{(\tau, 0)}(X \Rightarrow_1 [\varphi]) \& (\varphi \in K')\}$, $R_0^{(\nu)} \cap R_0^{(\mu)} = \varnothing$ для $\nu \neq \mu$. Аналогично соответствующие мнения представлены множествами $\Phi_0^{(\nu)} = \{\varphi | \exists X J_{\langle \nu, 0 \rangle}(X \Rightarrow_1 [\varphi]) \& (\varphi \in K')\}$, $\nu \in \{+1, -1, 0\}$, $\Phi_0^{(\tau)} = \{\varphi | \exists X J_{(\tau, 0)}(X \Rightarrow_1 [\varphi]) \& (\varphi \in K')\}$.

В случае, если выбор отношения к теме (и/или элементам каркаса) осуществляется на основе аргументов «за» и «против» (о соответствующей логике аргументации см. примеч. 24), множества мнений задаются непротиворечиво: $\forall \nu \forall \mu (\nu \neq \mu \supset \Phi_0^{(\nu)} \cap \Phi_0^{(\mu)} = \varnothing)$, $\nu, \mu \in \{+1, -1, 0, \tau\}$. Однако в общем случае это не обязательно. Для $\mu \neq \nu$ возможны также варианты: а) $\exists \nu \exists \mu (\Phi_0^{(\nu)} = \Phi_0^{(\mu)})$; б) $\exists \nu \exists \mu (\Phi_0^{(\mu)} \cap \Phi_0^{(\nu)} = \Phi_0^{(\nu)})$; в) $\exists \nu \exists \mu ((\Phi_0^{(\mu)} \cap \Phi_0^{(\nu)} \neq \varnothing) \& \neg ((\Phi_0^{(\mu)} \cap \Phi_0^{(\nu)} = \Phi_0^{(\nu)}) \vee (\Phi_0^{(\mu)} \cap \Phi_0^{(\nu)} = \Phi_0^{(\mu)})))$. Для всех этих случаев в процессе ДСМ-рассуждения могут быть порождены противоречивые гипотезы (разумеется, если сходства соответствующих респондентов также совпадут), поскольку (+)- и (–)-гипотезы являются в ДСМ-методе взаимно фальсифицирующими. Однако и схема *AlAnAb* может быть дополнена соответствующими процедурами.

Рассмотрим определенные выше множества мнений $\Phi_0^{(1)} = \{\varphi_1, \ldots, \varphi_{S_1}\}$, $\Phi_0^{(-1)} = \{\varphi_1, \ldots, \varphi_{S_2}\}$, $\Phi_0^{(0)} = \{\varphi_1, \ldots, \varphi_{S_3}\}$ и соответствующие множества респондентов $R_0^{(1)} = \{X_1, \ldots, X_{m_1}\}$, $R_0^{(-1)} = \{X_1, \ldots, X_{m_2}\}$, $R_0^{(0)} = \{X_1, \ldots, X_{m_3}\}$. Для каждой СДНФ $\varphi_1 \vee \ldots \vee \varphi_{S_i}$, $i = 1, 2, 3$, с помощью обобщенного алгоритма Куайна строятся сокращенные ДНФ $\partial(\varphi_1 \vee \ldots \vee \varphi_{S_i}) = \chi_1 \vee \ldots \vee \chi_{r_i}$ с соответствующим множеством импликант $[\partial\varphi]^{(\nu)} = \{\chi_1, \ldots, \chi_{r_i}\}$, $\nu \in \{+1, -1, 0\}$. Каждой импликанте χ_j из $[\partial\varphi]^{(\nu)}$ поставим в соответствие такое множество $\Phi_{0\,j}^{(\nu)}$ мнений φ, что φ покрывается импликантой χ_j, $\Phi_{0\,j}^{(\nu)} = \{\varphi \mid \chi_j \sqsubset \varphi\}$, $j = 1, \ldots, r_i$, $i = 1, 2, 3$. Соответственно, множество субъектов, мнение

которых есть элемент $\Phi_0{}^{(\nu)}{}_j$, обозначим $R_0{}^{(\nu)}{}_j = \{X| J_{\langle 1,0\rangle}(X\Rightarrow_1[\varphi_q]) \& \varphi_q \in \Phi_0{}^{(\nu)}{}_j\}$, $R_0{}^{(\nu)}{}_j = \{X_{j_1}, ..., X_{j_h}\}$. Сходство элементов $R_0{}^{(\nu)}{}_j$ – всех X таких, что их мнение покрывается импликантой χ_j, обозначим $V'^{(\nu)}{}_j$, $V'^{(\nu)}{}_j = \bigcap_{k=1}^{h} X_{j_k}$. Тогда отношение каузальности $C^{(\nu)}(V'^{(\nu)}{}_j, \chi_j)$ будет представлено парами $\langle V'^{(\nu)}{}_j, \chi_j\rangle$ ($V_j' \neq \varnothing$, $j = 1, ..., r_i$, $i = 1, 2, 3$). Итак, $V'^{(\nu)}{}_j$ – детерминанта мнения $\varphi_q \in \Phi_0{}^{(\nu)}{}_j$, $V'^{(\nu)}{}_j \subseteq X$, $J_{\langle \nu, 0\rangle}(X\Rightarrow_1[\varphi_q])$. Множеству импликант $[\partial\varphi]^{(\nu)}$ соответствует множество детерминант $\{V'^{(\nu)}{}_1, ..., V'^{\nu}_{r_i}\}$.

Абдуктивное объяснение исходных примеров из БФ, как и прежде, в этом случае имеет место всегда, если только сходство $V'^{(\nu)}{}_j$ респондентов, мнение которых (так же, как и в рассматриваемом примере) покрывается импликантой χ_j, не пусто. А вот предикат для вывода по аналогии для доопределения отношения к теме субъектов, описание которых представлено в БФ, может быть уточнен. Вследствие того что респонденты с разным отношением к теме могут иметь одинаковые мнения, возможны импликанты χ, входящие одновременно в разные множества $[\partial\varphi]^{(\nu)}$ и $[\partial\varphi]^{(\mu)}$ ($\mu \neq \nu$, $\nu, \mu \in \{+1, -1, 0\}$), и мнения, покрываемые импликантами для разных ν и μ одновременно. Следовательно, для порождения, например, гипотезы $J_{\langle 1,2\rangle}(X\Rightarrow_1 Y)$ необходимо проверить, выполняется ли (модифицированный) предикат $\Pi^+(X, Y) \rightleftharpoons \exists V'^{(+)}\exists\chi((V'^{(+)}\subseteq X)\&(V'^{(+)}\neq\varnothing)\&([\chi]\subseteq Y)\&C^{(+)}(V'^{(+)}, \chi)\&\forall\psi(([\psi]\subseteq Y)\rightarrow\neg \rightarrow\neg\ (\exists V'^{(-)}(C^{(-)}(V'^{(-)}, \psi)\&V'^{(-)}\subseteq X)\vee\exists V'^{(0)}(C^{(0)}(V'^{(0)}, \psi)\&V'^{(0)}\subseteq X)))$. $\Pi^-(X, Y)$ и $\Pi^0(X, Y)$ определяются аналогично.

Очевидно, что предложенный предикат позволяет осуществить более точное предсказание, чем это делалось в случае булевского опроса. Вследствие появления противоречий в этом случае может оказаться содержательным последовательное и циклическое применение процедур поиска сходства и вывода по аналогии, как это делается в ДСМ-методе. Однако и здесь в силу замкнутости мира при порождении импликант корректное доопределение возможно только для мнений из множества K', которые уже встречались в БФ.

Отметим, что все сказанное может быть распространено на вариант опроса, когда опрос по теме оказывается l-значным ($l \geq 5$), но это требует специального рассмотрения ДСМ-метода с более чем четырьмя типами истинностных значений (так, пример для пяти типов можно найти в литературе[26]).

Представленное сравнение вариантов формализованных когнитивных эвристик анализа эмпирических социологических данных демонстрирует различие в возможностях каждой из них. Алгебраический подход порождает меньшее число гипотез о причинах, которые при этом являются наиболее правдоподобными (мы говорим здесь о почти неформальном понимании правдоподобия), поскольку они являются максимальными по числу «родителей» – использовавшихся при их порождении примеров. Далее, в этом подходе степень абдуктивной объясненности – отношение числа объясненных примеров к общему числу примеров в исходной БФ – как правило, превышает аналогичный показатель для ДСМ-метода. Объясняется это тем, что в ДСМ-методе требуется объяснить *все* свойства объектов, тогда как в алгебраическом подходе достаточно покрытия имеющихся свойств соответствующими импликантами. И, наконец, к наиболее существенным различиям приводит вывод по аналогии: в алгебраическом подходе, в отличие от ДСМ-метода, невозможно формальное доопределение примеров с ранее не встречавшимися свойствами (мнениями).

Эти различия заставляют вновь обратиться к тезису о необходимости выбора для анализа эмпирических данных средств, адекватных природе задачи. Фундаментальным отличием ДСМ-метода автоматического порождения гипотез от алгебраического подхода является возможность его гибкого функционирования в открытом мире, тогда как алгебраический подход предполагает полноту описания исследуемой проблемы, онтологическую замкнутость. Кроме того, эвристика *AlAnAb* не апеллирует к содержательным соображениям, что, например, имеет место в случае запрета на контрпримеры в ДСМ-методе.

Полученные результаты могут послужить поводом к размышлению для сторонников применения формальных методов как таковых для анализа социологических данных. Но и ригористы, настаивающие на принципиальной невозможности анализа качественных социологических данных формальными средствами, возможно, согласятся смягчить свою точку зрения.

Реализация средств формализованного качественного анализа социологических данных в интеллектуальных системах повышает степень объективизации эмпирических социологических исследований и расширяет возможности решения задач когнитивной социологии – автоматического получения нового знания на основе эмпирических фактов.

Автор выражает благодарность проф. В.К. Финну за плодотворные обсуждения.

Примечания

[1] *Сорокин П.А.* Квантофрения // Социология. Хрестоматия для вузов / Сост. А.И. Кравченко. М.: Академический проект, 2002. С. 63–74.

[2] См.: *Татарова Г.Г.* Методы анализа данных в социологии. М.: Стратегия, 2002; *Готлиб А.С.* Введение в социологическое исследование (качественный и количественный подходы). М.: Флинта, 2005.

[3] См.: *Ядов В.А.* Стратегия социологического исследования. М.: Добросвет, 2003.

[4] См.: *Страусс А., Корбин Дж.* Основы качественного исследования. Обоснованная теория. Процедуры и техники. М.: КомКнига, 2007.

[5] *Fayyad U.M., Piatetsky-Shapiro G., Smyth P.* From Data Mining To Knowledge Discovery: An Overview // Advances in Knowledge Discovery and Data Mining / Ed. by U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy. AAAI Press/The MIT Press, Menlo Park, CA, 1996. P. 1–34.

[6] Мы не рассматриваем здесь методы анализа данных, объединенных общим названием *data mining* (см., например: *Чубукова И.А.* Data Mining. М.: Бином, 2008), некоторые из которых находят свое применение и в задачах анализа социологических данных. Эти методы, как правило, направлены не на извлечение закономерностей (знаний) из данных, а на решение задач классификации, кластеризации, управления и т. п.

[7] См.: *Ragin C.C.* The Comparative Method: Moving beyond Qualitative and Quantitative Strategies. Berkley, Los Angeles and London: University of California Press, 1987; См. также обзор: *Rihoux B.* Qualitative Comparative Analysis and Related Systematic Comparative Methods // International Sociology. 2006. Sept. Vol. 21 (5). P. 679–706.

[8] См.: *Ragin C.C.* Fuzzy-Set Social Science. Chicago: University of Chicago Press, 2000.

[9] *Финн В.К.* Правдоподобные рассуждения в интеллектуальных системах типа ДСМ // Автоматическое порождение гипотез в интеллектуальных системах. М.: Книжный дом «Либроком», 2009. С. 10–48.

[10] Автоматическое порождение гипотез в интеллектуальных системах. Ч. III. С. 409–491.

[11] Описание варианта такого рода ИС можно найти в: *Михеенкова М.А., Феофанова Т.Л.* Обучающая ДСМ-система для анализа социологических данных // Вестник РГГУ. 2009. № 10. Серия «Информатика. Информационная безопасность. Математика». С. 152–169.

[12] *Финн В.К.* Синтез познавательных процедур и проблема индукции // Автоматическое порождение гипотез в интеллектуальных системах. С. 92–158.

[13] *Михеенкова М.А., Финн В.К.* Правдоподобные рассуждения и булева алгебра для анализа социологических данных (проблемы когнитивной

социологии) // Математическое моделирование социальных процессов. М.: Университет; Книжный дом, 2009. Вып. 10. С. 229–236. См. также: *Финн В.К., Михеенкова М.А., Сидорова А.В.* О когнитивных эвристиках анализа социологических данных // III Всероссийский социологический конгресс «Социология и общество: проблемы и пути взаимодействия», Москва, 21–24 октября 2008 г.: Тезисы докладов и выступлений. URL: http://www.isras.ru/abstract_bank/1214905899.pdf.

[14] *Парсонс Т.* О теории и метатеории // Теоретическая социология. Антология. М.: Наука, 2002. Т. 2. С. 44–45.

[15] См.: *Михеенкова М.А.* О принципах формализованного качественного анализа социологических данных // Информационные технологии и вычислительные системы. 2009. № 4. (в печати).

[16] *Финн В.К.* Правдоподобные рассуждения в интеллектуальных системах типа ДСМ. С. 54–101.

[17] *Финн В.К.* Автоматическое порождение гипотез в интеллектуальных системах. С. 446–484.

[18] *Финн В.К., Михеенкова М.А.* К формальному определению закрытого социологического опроса // Сорокинские чтения «Социальные процессы в современной России: традиции и инновации», Москва, 4–5 декабря 2007 г.: В 5 т. М.: Университет; Книжный дом, 2007. Т. 1. С. 214–217.

[19] См.: *Финн В.К.* Логика качественного анализа социологических данных. М., 2009 (в печати).

[20] См.: *Финн В.К.* Правдоподобные рассуждения в интеллектуальных системах типа ДСМ.

[21] *Финн В.К.* Автоматическое порождение гипотез в интеллектуальных системах. С. 446–484.

[22] *Финн В.К., Михеенкова М.А.* К формальному определению закрытого социологического опроса // Сорокинские чтения. 2007. Т. 1. С. 214–217.

[23] *Финн В.К.* Автоматическое порождение гипотез в интеллектуальных системах. С. 485–491.

[24] *Финн В.К.* Стандартные и нестандартные логики аргументации // Многозначные логики и их применения. М.: ЛКИ, 2008. Т. 2: Логики в системах искусственного интеллекта / Под ред. В.К. Финна. С. 59–91.

[25] *Яблонский С.В.* Введение в дискретную математику. М.: Наука, 1986. С. 297–336.

[26] *Финн В.К., Михеенкова М.А.* О логических средствах концептуализации анализа мнений // Многозначные логики и их применения. Т. 2. С. 152–199.

Ю.В. Козлова

ОБ ОБЪЕКТНОЙ МОДЕЛИ СИСТЕМЫ ТРЕБОВАНИЙ ДЛЯ ГЕНЕРАТОРА ТЕСТОВЫХ ПРИМЕРОВ «TestJSM!»

Описывается модифицированная объектная модель системы требований для программы «TestJSM!», которая порождает тестовые данные для демонстрации особенностей различных вариантов ДСМ-метода.

Ключевые слова: ДСМ-метод, ДСМ-система, варианты ДСМ-метода, тестирование, тестовые данные, генератор тестовых примеров, объектная модель, система требований к тестовым примерам.

В статье рассматривается новая объектная модель системы требований для генератора «TestJSM!», рассмотренного автором в статье «Генератор тестовых примеров для различных вариантов ДСМ-метода»[1]. Программа «TestJSM!» порождает тестовые примеры, демонстрирующие возможности разных вариантов ДСМ-метода. Кроме того, эта программа может быть использована для проверки стратегий ДСМ-метода[2], а также для сравнения выразительной силы ДСМ-стратегий[3]. Программа написана на языке Visual Prolog.

Первый вариант объектной модели системы требований, описанной автором[4], изображен на рис. 1. Объектная модель включает коллекции и объекты. Названия коллекций находятся в прямоугольниках с толстой рамкой, а названия объектов – в прямоугольниках с тонкой рамкой.

Особенностями работы программы «TestJSM!», разработанной на основе данного варианта объектной модели, являются:

– порождение тестовых примеров, демонстрирующих итерации применения правил только простого ДСМ-метода;

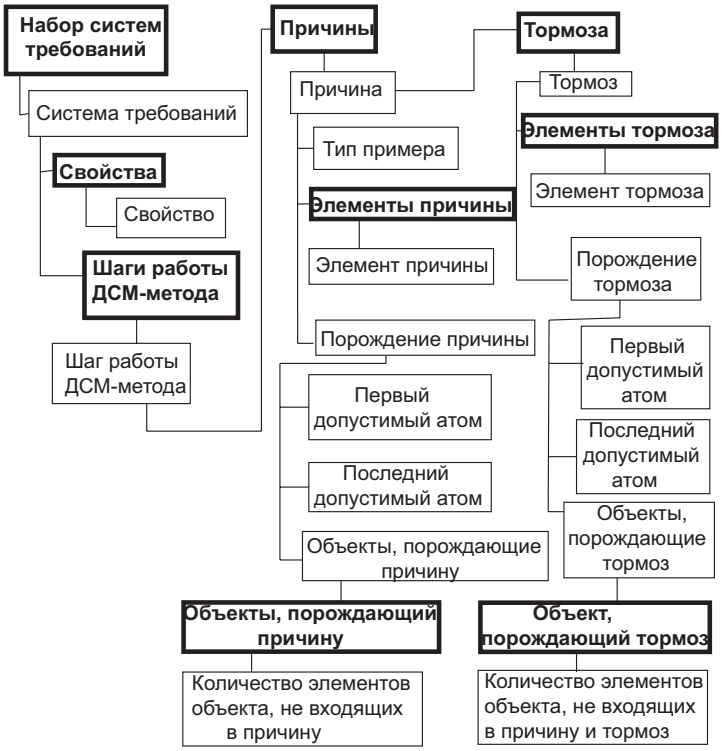


Рис. 1. Первый вариант объектной модели системы требований

– явное указание количества и структуры неопределенных примеров;
– ограничения на объекты, порождающие причины и тормоза, указываются для каждой причины (тормоза).

Настройка параметров требований, которые описываются согласно данному варианту объектной модели, осуществляется с помощью мастера, который последовательно предлагает пользователю:

– выбор ДСМ-стратегии;
– ввод желаемых причин, свойств, тормозов;
– определений ограничений на объекты, порождающие причины и тормоза.

Таким образом, пользователю приходится в явном виде обозначать, для какого варианта ДСМ-метода будет осуществляться ввод требований.

Модифицированная объектная модель системы требований

Для генератора «TestJSM!» был разработан новый вариант объектной модели, который предоставляет пользователю новые возможности по работе с программой и позволяет получать новые виды тестовых примеров.

Объектная модель включает коллекции и объекты. На диаграмме объектной модели названия коллекций находятся в прямоугольниках с толстой рамкой, а названия объектов – в прямоугольниках с тонкой рамкой. В структуру объектной модели входят следующие компоненты.

1. Коллекция «Набор систем требований» – самый верхний компонент в иерархии объектной модели, который включает в себя все остальные компоненты (рис. 2). Каждый объект «Система требований» (или «Задача») данной коллекции включает в себя следующие компоненты:

a) объект «Название задачи»,

b) объект «Назначение задачи»,

c) коллекцию «Атомы»,

d) объект «Формирование причины»: пользователь указывает для данной задачи, как формируется причина (причина является множеством элементов, полученных пересечением объектов, либо причина является частью пересечения объектов),

e) коллекцию «Свойства» (рис. 3),

f) коллекцию «Дополнительные атомы»,

g) объект «Минимальное количество дополнительных атомов»,

h) объект «Максимальное количество дополнительных атомов»,

i) объект «Максимальное количество неопределенных объектов»: максимальное количество объектов, о которых не известно, как они связаны с целевыми свойствами,

j) объект «Минимальное количество примеров»: минимальное количество примеров, при пересечении которых получаются введенные пользователем «+»- и «–»-причины,

2. Коллекция «Атомы» включает в себя элементы, из которых состоят «+»- и «–»-причины.

3. Коллекция «Дополнительные атомы» включает в себя элементы, которые не входят в «+»-причины и «–»-причины, введенные пользователем.

4. Коллекция «Свойства» – названия свойств, для которых указываются причины наличия/отсутствия свойств или информация о том, что данных о связи свойств и фрагмента нет; причем предполагается, что соответствующая гипотеза выдвигается не раньше,

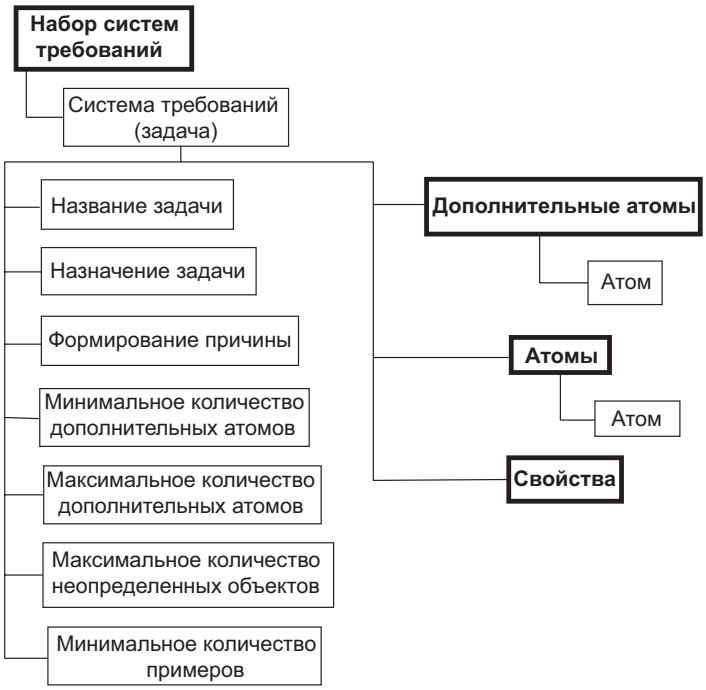


Рис. 2. Коллекция «Набор систем требований»

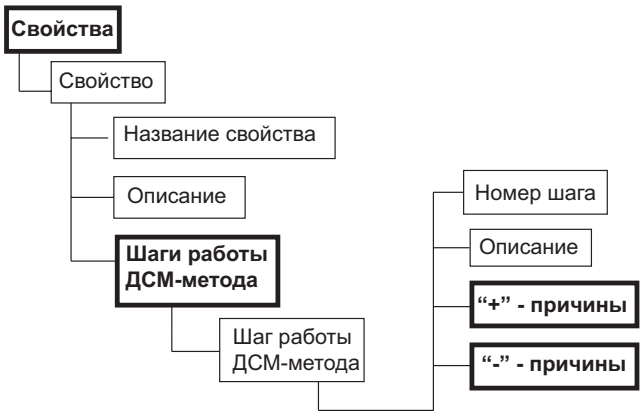


Рис. 3. Коллекция «Свойства»

чем на n-м шаге работы ДСМ-метода (n⩾1). Каждый объект «Свойство» данной коллекции включает в себя:

a) объект «Название свойства»,

b) объект «Описание» – дополнительная информация вводится в свободном формате,

c) коллекцию «Шаги работы ДСМ-метода».

5. Коллекция «Шаги работы ДСМ-метода». Предполагается, что на *n*-м шаге работы ДСМ-метода во время процедуры индукции выдвигается гипотеза о возможных причинах наличия или отсутствия целевых свойств у объектов. Каждый объект «Шаг работы ДСМ-метода» включает в себя:

a) объект «Номер шага»,

b) объект «Описание» – дополнительная информация вводится в свободном формате,

c) коллекцию «+»-причины (рис. 4),

d) коллекцию «–»-причины.

6. Коллекция «+»-причины.

7. Коллекция «–»-причины.

Каждый объект коллекций «+»-причины и «–»-причины включает в себя следующие компоненты:

a) объект «Название причины»,

b) коллекцию «Атомы»,

c) коллекцию «Тормоза».

8. Коллекция «Атомы» включает в себя элементы структуры причины.

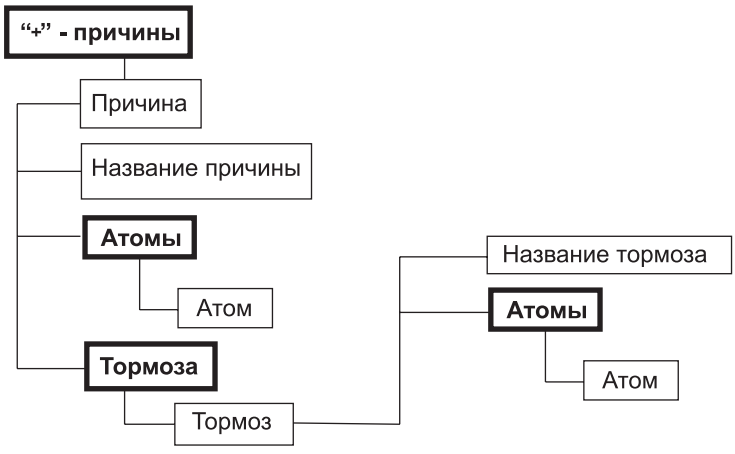


Рис. 4. Коллекция «+»-причины

9. Коллекция «Тормоза» содержит информацию о возможных тормозах данной «+»- или «–»-причины свойства. Каждый объект данной коллекции содержит:

a) объект «Название тормоза»,

b) коллекцию «Атомы».

10. Коллекция «Атомы» включает в себя элементы структуры тормоза.

На рис. 5 представлена структура новой объектной модели системы требований.

Так же как и в первом варианте объектной модели, при разработке нового варианта объектной модели учитывались все особенности известных ДСМ-стратегий.

Отличительной особенностью работы программы «TestJSM!», разработанной на основе нового варианта объектной модели, является порождение тестовых примеров, демонстрирующих итерации применения правил как простого ДСМ-метода, так и обобщенного и несимметричного ДСМ-метода, а также модифицированного ДСМ-метода[5].

Кроме того, программа, разработанная на основе нового варианта объектной модели, предоставляет пользователю следующие возможности:

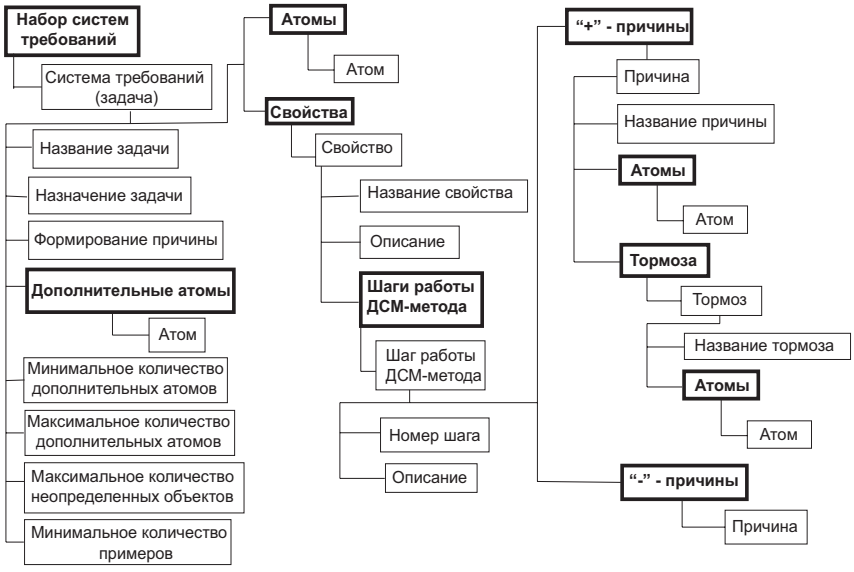


Рис. 5. Новый вариант объектной модели системы требований

- пользователь не указывает структуру неопределенных примеров, а указывает лишь максимальное количество неопределенных примеров;
- пользователь указывает общие сведения по ограничению на объекты, порождающие причины и тормоза, применимые для всех причин (тормозов);
- пользователю не нужно в явном виде обозначать, для какого варианта ДСМ-метода будет осуществляться ввод требований.

Итак, в работе была описана модифицированная объектная модель системы основных требований к тестовым примерам, порождаемым генератором «TestJSM!». Также были описаны основные отличительные особенности работы генератора, разработанного на основе нового варианта объектной модели.

Примечания

1 См. в настоящем номере: *Козлова Ю.В.* Генератор тестовых примеров для различных вариантов ДСМ-метода.

2 См.: *Финн В.К.* Правдоподобные рассуждения в интеллектуальных системах типа ДСМ // Итоги науки и техники. Сер. «Информатика». 1991. Т. 15: Интеллектуальные информационные системы.

3 См.: *Козлова Ю.В.* Указ. соч.

4 Там же.

5 См.: *Аншаков О.М.* Об одном подходе к порождению гипотез в ДСМ-методе // Десятая национальная конференция по искусственному интеллекту с международным участием КИИ-2006 (Обнинск, 25–28 сент. 2006 г.): Труды конференции: В 3 т. М., 2006.

А.Н. Приезжая

АВТОМАТИЗИРОВАННАЯ РАЗРАБОТКА ЗАЩИЩЕННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Процесс разработки информационной системы в защищенном исполнении требует трудоемкого анализа защищаемого объекта и разработки большого количества документации. В данной статье предлагается технология, основанная на автоматизированном преобразовании UML-моделей, которая позволяет одновременно разрабатывать несколько представлений объекта, включая текстовое описание, и в автоматизированном режиме строить на их основе модели угроз и нарушителя, а также модель системы защиты. В данной статье разработка защищенной информационной системы рассматривается на примере распределенной базы данных.

Ключевые слова: UML-модель, персональные данные, преобразование моделей, модель объекта, модель угроз, документация, модель системы в защищенном исполнении.

В России существует огромное количество видов (категорий) информации, подлежащей защите (по некоторым оценкам до 30), и к каждому из них предъявляются свои требования по обеспечению безопасности[1], кроме того, в связи с развитием информационных технологий постоянно растет сложность систем обработки данных. В результате специалисты в области информационной безопасности сталкиваются с необходимостью защищать все более сложные системы в соответствии с разными, порой противоречивыми, требованиями, и при этом сроки разработки СЗИ крайне ограничены[2]. В связи со всем вышесказанным особую актуальность приобретает создание средств поддержки разработки защищенных систем.

Предложенный автором метод автоматизированной разработки информационных систем в защищенном исполнении включает в

себя разработку нескольких взаимосвязанных формальных моделей объекта, предназначенных для автоматизированного преобразования, построенного с использованием технологии MDA, в результате которого генерируются модель угроз безопасности информации, модель системы защиты и модель защищенной информационной системы. Рассматриваемый подход упрощает анализ системы, сокращает затраты времени на рутинные процедуры и снижает вероятность ошибки, иными словами, снижает финансовые и временные затраты на разработку защищенной системы.

В данной статье процесс разработки ИС в защищенном исполнении рассматривается на примере распределенной системы, обрабатывающей персональные данные. Взаимодействие распределенных компонент системы осуществляется через каналы общего пользования, то есть для защиты передаваемой информации конфиденциального характера необходимо применение СКЗИ.

В зарубежной научной литературе разработано несколько методов преобразования моделей с использованием технологии MDA. В частности, предложен ряд диалектов UML для разработки систем безопасности[3]. Также в ряде статей рассматривается применение в интегрированной разработке систем в защищенном исполнении технологии MDA для какого-то одного аспекта (механизма) безопасности[4].

Существующие на сегодняшний день технологии предусматривают разработку СЗИ параллельно с функциональной частью системы, тогда как применение UML и MDA для разработки СЗИ на этапе эксплуатации системы в литературе не разработано. Кроме того, существующие методы автоматизированной разработки предусматривают только внедрение функций безопасности в информационную систему, в то время как метод, предложенный автором, предусматривает автоматизацию работ, выполняемых на предшествующих этапах, а именно разработку моделей угроз и потенциальных возможностей нарушителя, соответствующую требованиям российского законодательства.

Общие принципы разработки информационной системы в защищенном исполнении

Построение системы безопасности корпоративной информационной системы (ИС) невозможно без четкого представления о том, как она фактически функционирует, а также о том, в какой мере и каким образом ее ресурсы уже защищены. Для решения этой задачи проводится изучение информационной системы и ее компонент,

а также организации работы пользователей с целью выявления основных информационных ресурсов и информационных потоков, их соотнесения с различными категориями обрабатываемой информации, определяются границы системы, для которой должен быть обеспечен режим информационной безопасности. Соответственно определяется существующая структура организации, размещение средств вычислительной техники и поддерживающей инфраструктуры, технология обработки информации, ресурсы информационной системы, подлежащие защите. Для оценки ресурсов выбирается система критериев и методология получения оценок по этим критериям.

На основе полученных на этапе предпроектного обследования данных строится модель защищаемого объекта. В сложившейся практике модель объекта, как правило, представляется в форме вербального описания объекта защиты. В рамках данной статьи используется модель объекта, построенная на формальном языке моделирования (UML-модель), так как создание текстового описания объекта по его формальной модели является достаточно простой задачей, а сама формальная модель объекта делает объект более обозримым для разработчика, позволяя рассмотреть систему с разных точек зрения, с заданным уровнем детализации. Кроме того UML-модель объекта используется в ходе автоматизированной разработки модели информационной системы в защищенном исполнении.

На основе модели объекта разрабатываются модели угроз безопасности информации и вероятного нарушителя, проводится анализ рисков и разрабатывается перечень актуальных угроз. Также в ходе данного этапа разработки создаются политики безопасности и частное техническое задание на разработку СЗИ.

Техническое проектирование ИС в защищенном исполнении включает разработку технического проекта (пояснительной записки, ведомости технического проекта, плана организационно-технических мероприятий по подготовке ИС к внедрению средств и мер защиты информации и т. д.). На данном этапе определяются конкретные технические решения, реализующие выбранные политики безопасности и разработанную для данной автоматизированной системы концепцию СЗИ.

На следующем этапе разрабатываются положения по организации и проведению работ по обеспечению безопасности информации при ее обработке в ИС, требования по обеспечению безопасности информации при ее обработке в ИС, инструкции персоналу в части обеспечения безопасности информации при ее обработке в ИС и т. д. В случае необходимости проводятся аттестационные испытания ИС по требованиям безопасности информации.

Предложенный в данной работе метод разработки информационной системы в защищенном исполнении предназначен в первую очередь для автоматизации работ, проводимых на предпроектном этапе и частично на этапе технического проектирования. В ходе этого этапа не только проводится первичный анализ объекта защиты, определение его границ и основных свойств, но и разрабатываются документы, определяющие принципы построения системы защиты. При этом необходимо учитывать, что разрабатываемая интеграторами система защиты информации и сопровождающая ее документация должны соответствовать требованиям регуляторов в области обеспечения безопасности информации. В качестве регуляторов в данной области в настоящий момент выступают ФСТЭК России и ФСБ России[5]. Каждая из структур разрабатывает собственную обязательную для выполнения нормативно-методическую базу, кроме того, НМД привязана к виду защищаемой информации и типу ИС.

При этом процессы обработки информации организованы таким образом, что выделение персональных данных из общего массива информации конфиденциального характера нецелесообразно, а порой и невозможно. Таким образом, автоматизированные системы одновременно обрабатывают информации различных видов: коммерческая тайна и персональные данные (ПДн), служебная тайна и персональные данные и т. д. В этом случае может возникнуть необходимость совмещения требований регуляторов к режиму защиты различных видов информации, также в случае принятия решения об использовании криптографических (шифровальных) средств для обеспечения безопасности персональных данных возникает необходимость совмещения требований ФСТЭК России и ФСБ России как в части разработки документов, так и в части предъявляемых требований, что далеко не всегда является простой задачей.

Разработка модели объекта

Модель угроз и модель нарушителя, а впоследствии и система защиты информации строятся применительно к конкретному объекту с учетом особенностей его функционирования. Следовательно, для построения грамотной модели угроз безопасности и возможного нарушителя необходимо провести анализ объекта.

Анализ объекта может быть проведен с применением различных методик. Одной из возможных методик анализа является моделирование, которое позволяет получить обозримое и полное представление системы с требуемым уровнем детализации, кроме того, в рамках предложенного автором подхода модель объекта ис-

пользуется для автоматизированного получения модели угроз и в дальнейшем построения модели защиты.

Модель объекта должна быть построена максимально полной, с различными представлениями объекта и различными уровнями детализации, так как информация, которая может показаться избыточной на одном этапе, может быть использована на последующих. Например, данные об используемых протоколах не нужны для определения принципов построения системы защиты, но могут оказать влияние на выбор технических средств.

Для построения UML-модели объекта необходимы следующие исходные данные:

- *перечень информации*, подлежащей защите;
- *расположение АС* относительно границ контролируемой зоны (КЗ);
- *конфигурация и топология АС* в целом и ее отдельных компонент, в том числе физические, функциональные и технологические связи как внутри этой АС, так и с другими системами различного уровня и назначения;
- *технические средства и системы*, использующиеся в АС, условия их расположения;
- *общесистемные и прикладные программные средства*, имеющиеся или предлагаемые к использованию (разработке) в АС;
- *режимы обработки* информации в АС в целом и в ее отдельных компонентах;
- *степень участия персонала* в обработке информации конфиденциального характера, характер его взаимодействия между собой.

На основании полученных исходных данных строится модель объекта, включающая описание:

- защищаемых ресурсов (информация, программное обеспечение, технические средства, средства защиты и т. п.);
- информации, обрабатываемой в системе (класс, количество, формы представления и др.);
- потоков информации;
- подключения к сетям общего пользования;
- архитектуры системы (логика ее построения);
- функционального взаимодействия компонентов системы;
- размещения (с точки зрения топологии сети, программных модулей);
- технических средств (аппаратные и программные средства, используемые протоколы, порты);
- размещения ТС в границах контролируемой зоны;
- пользователей системы (должностные обязанности, режим доступа, права доступа);

– существующих мер и средств защиты (в том числе резервное копирование) в той мере, в которой они будут сохранены в информационной системе в защищенном исполнении.

Результаты информационного обследования объекта, как правило, оформляются в виде текстового отчета, однако в рамках данной работы в качестве описания объекта будет рассматриваться формальная модель на языке UML. Данная модель не только упростит процесс анализа объекта для разработчика, сделав объект более обозримым, но и позволит ему создавать модель угроз безопасности и потенциального нарушителя, а затем и модель защиты в автоматизированном режиме, который позволяет контролировать соответствие разрабатываемых моделей исходным данным (собственно модели объекта) и требованиям регуляторов.

Модель объекта формируется из двух основных представлений: модели бизнес-логики системы и модели реализации. Модель бизнес-логики содержит описание технологического процесса в форме сценариев UML и диаграмм классов, отражающих логику построения, функциональные подсистемы, основные информационные потоки, штатную структуру организации. Модель реализации содержит описание объектов доступа, размещения компонент системы и их спецификации, информационного обмена между распределенными частями ИС, а также диаграммы действия, описывающие реализацию сценариев техническими средствами системы, субъектов доступа системы и их прав и т. д. Все представления системы взаимосвязаны, так как они отражают один и тот же объект с разных точек зрения и с различной степенью детализации.

Для большей наглядности проиллюстрируем создание модели объекта описанием распределенной базы данных. В рассматриваемом примере ИС имеет три уровня: центральный, региональный и уровень пользователя. Информация вводится в региональные базы, затем по каналу связи передается в центральную базу данных, удаленные пользователи посылают запросы в региональную или центральную базу. База данных состоит из двух независимых систем обработки запросов (обусловленных формой представления запрашиваемой информации–изображение или текст). Взаимодействие между системами осуществляется по открытому каналу связи.

Рассмотрим порядок построения модели объекта. Одним из основных представлений объекта является диаграмма классов, описывающая логическую структуру ИС; в нашем примере данная диаграмма включает в себя объекты центрального, регионального уровней и уровня пользователей и отражает связи между этими объектами. Каждый тип объекта представляется классом, уровень объекта определяется стереотипом, а его тип отражается в имени класса (рис. 1).

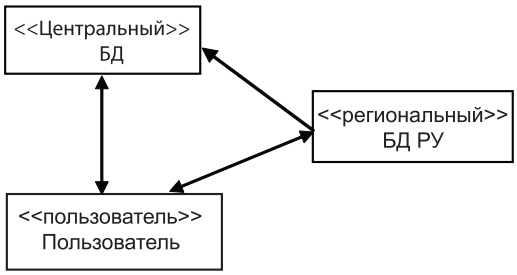


Рис. 1. Логическая структура – первое приближение

Взаимодействия между объектами отображаются двумя способами: с помощью направленных ассоциаций (в свойствах ассоциации содержатся данные об информации, передающейся между уровнями) и на диаграммах последовательности, где взаимодействие объектов моделируется более подробно. Диаграммы последовательности строятся для основных сценариев и отображают обмен информацией между объектами разного уровня. Еще одна форма представления – диаграмма классов, описывающая функциональные подсистемы; данный вид диаграммы используется в тех случаях, когда объекты имеют выраженное разделение по функциональному признаку, отличное от его логической структуры.

В общем случае объект каждого уровня может представлять собой сложную распределенную ИС со своей логической структурой. В этом случае для каждого класса строится ассоциированный пакет, описывающий его внутреннюю логическую структуру. В нашем случае такие пакеты строятся для двух верхних уровней, при этом данный пакет имеет аналогичную структуру, то есть включает в себя классы, отображающие типы объектов и диаграммы классов, описывающие взаимодействия внутри объекта и – в случае необходимости – с объектами других уровней (при этом диаграммы последовательностей данного уровня детализируют взаимодействия между уровнями). Для сложных распределенных систем также выделяются функциональные подсистемы и описывается информационное взаимодействие между ними. Так, в рассматриваемом примере выделяются две подсистемы – индексации и поиска информации (в зависимости от формы представления информации) и коммуникационная подсистема.

Каждая функциональная подсистема представляется отдельным классом, взаимодействие между классами описывают диаграммы последовательности. В случае необходимости дальнейшей дета-

лизации для функционального класса создается ассоциированный пакет, отражающий его внутреннюю структуру.

Для определения угроз безопасности информации необходимо определить условия размещения компонент ИС, каналы связи и организационно-режимные мероприятия. Для полного описания условий функционирования объекта защиты строятся диаграмма (диаграммы) размещения логических компонент системы на серверах и узлах сети; диаграмма (диаграммы) размещения функциональных компонент системы на серверах и узлах сети.

Технические средства серверов и узлов сети моделируются классами со стереотипом ТС, каждый такой класс обладает атрибутами, отражающими перечень аппаратных компонент и установленного программного обеспечения. При этом для каждого технического средства или типа технического средства определяются условия размещения, организационно-режимные мероприятия, в частности режим доступа, контролируемая зона и т. д. Данные свойства задаются в специальном меню – Deployment, определяемом в файле deployment.pty. Также для технического средства могут быть определены характеристики безопасности[6], например неотказуемость. Аналогичным образом моделируется программное обеспечение. Диаграммы размещения технических средств определяют применение тех или иных ТС (ПО) на определенных логических и функциональных подсистемах.

При определении связи между распределенными компонентами необходимо определить свойства канала: является ли этот канал внутренним, выделенным, защищенным или открытым каналом общего пользования. Свойства канала отражаются в его стереотипе. Так, в рассматриваемом примере канал между региональным и центральным уровнями имеет стереотип «Internet», а канал, связывающий серверные компоненты регионального уровня, – «Secure LAN».

Необходимо определить информацию, обрабатываемую в системе. Для информационных ресурсов используются следующие представления:

– диаграмма классов, содержащая виды информации (персональные данные, коммерческая тайна и т. п.), обрабатываемые в системе. Каждый вид отображается отдельным классом, характеристики безопасности по умолчанию для данного вида информации задаются значениями соответствующих атрибутов этого класса (имя атрибута – характеристика безопасности, значение 0 не предъявляется, 1 предъявляется), например, к персональным данным по умолчанию предъявляются целостность и конфиденциальность, в процессе оп-

ределения характеристик безопасности конкретного информационного ресурса значения могут быть переопределены;
- диаграмма классов, отображающая информационные ресурсы автоматизированной системы, например фотоизображения, или паспортные данные, или данные банковского счета и т. п. Защищаемый информационный ресурс представляется отдельным классом, для каждого такого ресурса ассоциируется класс, представляющий вид информации, атрибуты этого класса содержат дополнительные характеристики безопасности или переопределяют характеристики по умолчанию. Кроме того, для классов, определенных как персональные данные, необходимо определить дополнительные атрибуты – категорию и количество;
- строятся диаграммы размещения информационных ресурсов на компонентах распределенной системы, а также по функциональным подсистемам. При этом для каждого компонента системы определяются формы, в которых на нем может существовать информация, и при определении информационного ресурса для него задаются возможные формы представления на данном сервере, узле – техническом средстве;
- строятся диаграммы взаимодействия, описывающие информационные потоки между компонентами и подсистемами ИС.

Строится диаграмма классов, описывающая структуру зарегистрированных пользователей системы, то есть субъектов доступа. В нашем примере это пользователь, оператор и системный администратор. Для удобства дальнейшего анализа вводится класс (группа), объединяющий пользователей (классы со стереотипом SystemUser).

Также строятся диаграммы, описывающие категории лиц, допущенных к системе, но не являющихся ее зарегистрированными пользователями. Эти классы объединяются общим стереотипом NonSystemUser. В качестве таких лиц могут рассматриваться:
- представители сторонних организаций;
- лица, обеспечивающие поставку и ремонт технических средств;
- представители организаций, взаимодействующих по вопросам обеспечения жизнедеятельности АС;
- обслуживающий персонал, производящий работы в помещениях, в которых размещается оборудование ИС (например, работники хозяйственных служб);
- другие сотрудники организации, имеющие доступ в помещения, в которых размешается оборудование ИС;
- и другие категории лиц.

Для всех категорий лиц, допущенных к системе, описываются существующие режимные ограничения, в том числе порядок допуска в контролируемую зону, возможность удаленной работы, порядок аутентификации и т. д. Данные ограничения задаются значениями соответствующих атрибутов класса.

Для каждой категории пользователей в системе (в нашем случае это администратор, пользователь и оператор) определяются полномочия. В первом приближении полномочия пользователя определяет сценарий. Также могут быть построены сценарии взаимодействия с системой лиц, не являющихся зарегистрированными пользователями системы.

Для данной системы необходимо создать несколько сценариев (Use case), например:

- ввод информации в БД;
- удаление информации из БД;
- поиск информации;
- синхронизация баз данных центрального и регионального уровней;
- порядок осуществления технической поддержки (осуществляется организацией-разработчиком удаленно);
- управление пользователями.

Естественно, каждый сценарий имеет несколько вариантов развития, в частности ввод новой информации в БД и попытка повторного ввода информации в БД. Сценарии необходимы для понимания функциональных задач системы и используются в дальнейшем для построения диаграмм последовательности, описывающих взаимосвязь компонент ИС и информационный обмен между ними.

Диаграмма последовательности представляет собой описание реализации сценария через обмен сообщениями (вызовами) между классами системы (рис. 2).

На основании UML-модели объекта строится текстовый шаблон описания объекта, который после доработки может быть включен в отчетную документацию. На основании практического опыта автором был разработан шаблон описания объекта – типовые разделы и принципы построения документа, для которых на основании модели с помощью скрипта (genDescription) автоматически генерируются текстовые описания. В общем случае полученный шаблон модели объекта имеет следующую структуру:

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ
ПРИНЯТЫЕ СОКРАЩЕНИЯ
1. ВВЕДЕНИЕ
2. ХАРАКТЕРИСТИКИ ОБЪЕКТА ЗАЩИТЫ
2.1. Основание для разработки

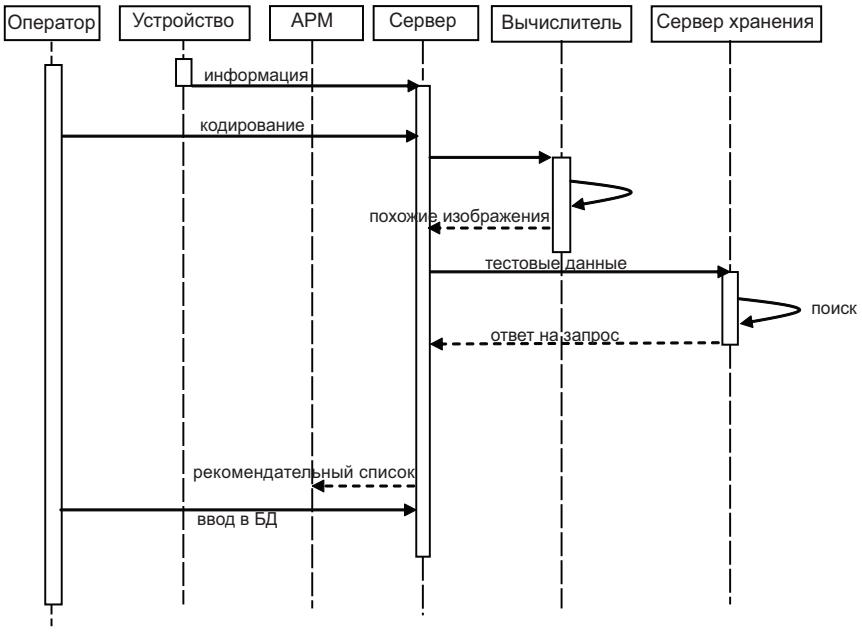


Рис. 2. Диаграмма поиска

2.2. Основания обработки персональных данных
2.3. Назначение, архитектура и структура
2.4. Состав, обрабатываемая информация и пользователи
2.5. Характеристика объекта защиты по уровням логической структуры
2.5.1. Общее описание
2.5.2. Архитектура
2.5.3. Функциональные возможности
2.5.4. Расположение
2.5.5. Состав и назначение технических средств
2.5.6. Информационное взаимодействие подсистем
2.5.7. Информационное взаимодействие уровней
2.6. Организационно-режимные меры
2.7. Классификация
2.8. Объекты защиты и защищаемые ресурсы

Разумеется, полученный текст не является готовым документом и требует доработки специалистом, однако наличие подобного шаблона документа и схем, его иллюстрирующих, значительно ускоря-

ет процесс документирования работ по созданию СЗИ и уменьшает вероятность появления расхождений между реальным состоянием системы и ее описанием в документации.

Разработка модели угроз и возможностей нарушителя

Модель угроз строится на основании модели объекта, отражающей особенности его функционирования, типовой модели угроз и экспертных решений. Предлагаемый инструмент разработки автоматизирует процесс анализа модели объекта и построения моделей угроз и нарушителя, в процессе разработки используются типовые модели, построенные в соответствии с руководящими документами ФСТЭК России и ФСБ России и основанные на практическом опыте автора[7].

Инструмент поддержки разработки модели угроз и модели потенциального нарушителя включает в себя:

1. Базовую модель угроз и потенциального нарушителя (Base.mdl)[8], которая содержит описание типового нарушителя, его возможностей, типовых уязвимостей программных и технических средств, а также модели каналов и способов реализации угроз. Кроме того, в данном пакете моделируются такие ключевые абстракции, как характеристики безопасности.

2. Шаблон модели объекта (некоторый набор правил, в соответствии с которыми строится модель объекта; описание построения модели объекта приведено в разд. 4).

3. Инструмент разметки базовой модели, позволяющий выбирать или исключать ее составляющие.

4. Инструмент разметки модели объекта, позволяющий задать требования для различных элементов модели.

В соответствии с методикой разработки при создании модели угроз безопасности информации и потенциального нарушителя необходимо:

– создать модель угроз верхнего уровня;

– создать модель нарушителя, то есть определить:

a) категории лиц – потенциальных нарушителей;

b) степень их осведомленности о системе;

c) перечень имеющихся в их распоряжении методов и средств проведения атак;

d) перечень возможных каналов атак;

– определить актуальные способы реализации, то есть детализированные угрозы.

Построенная на подготовительном этапе модель объекта позволяет в автоматизированном режиме построить модель угроз верхнего уровня, то есть выделить все ресурсы и связанные с ними характеристики безопасности. Инструмент преобразования (скрипт GenModelhigh) создает отдельную диаграмму для каждого типа угроз (например, угрозы модификации, угрозы нарушения конфиденциальности и т. п.), на которых отображаются объекты, этим угрозам подверженные, также данная модель представляется в текстовом виде.

Модель нарушителя строится экспертным путем на основании базовой модели нарушителя Base.mdl и диаграмм субъектов доступа. Диаграммы субъектов доступа используются для определения категорий лиц, которые могут рассматриваться в качестве нарушителя, и лиц, исключенных в силу организационных мер, из числа нарушителей. В зависимости от организационных ограничений, связанных с каждым классом субъекта доступа, определяется, к какому типу нарушителя (внутренний или внешний) относится данная категория лиц. Предположения о возможностях потенциального нарушителя и его знаниях о системе строятся с использованием предположений базовой модели (в частности, об уровне знаний и вооруженности типовых категорий нарушителя и перечень возможных каналов атак) на основании экспертной оценки, а также таких элементов модели объекта, как права доступа субъектов доступа – потенциальных нарушителей, и режим доступа, связанный с объектами защиты – техническими средствами, и свойства каналов связи. Так, инструмент преобразования автоматически определяет перечень объектов, к которым возможен удаленный доступ, а также прямой доступ внешнего нарушителя (например, компьютеры мобильных пользователей). Полученные данные анализируются экспертом, из рассмотрения могут быть исключены неактуальные каналы атак или переопределены возможности нарушителя.

На основании полученной модели нарушителя строится детализированная модель угроз. Для каждой угрозы безопасности верхнего уровня вида <ресурс> – нарушение <характеристики безопасности> – определяется возможный канал атаки и способ реализации угроз (то есть нарушения характеристики безопасности). Кроме того, ресурс связан с понятием уязвимости, наличие которой в свою очередь определяет возможность применения конкретного способа реализации. Для каждого типового канала доступа в базовой модели определены способы реализации угроз (то есть способы нарушения характеристик безопасности; так, например, для электромагнитного канала рассматривается способ нарушения конфиденциальности – перехват ПЭМИ, ВЧН и т. д.). Для каждого

способа реализации определен требуемый тип нарушителя и необходимая техническая вооруженность. Таким образом, в построении детализированной модели угроз задействованы следующие абстракции (см. рис. 3).

Каждая из представленных на схеме абстракций в модели представлена стереотипом класса, описывающего конкретный канал или, например, тип нарушителя.

Детализированная модель угроз имеет следующую структуру: для каждого канала доступа создается отдельный пакет, в котором создаются диаграммы для каждого ресурса, содержащие классы, описывающие способ реализации угрозы безопасности. Угрозы безопасности формируются не для каждого отдельного ресурса, а для типа ресурса, обладающего одинаковыми свойствами. Так, например, в рассматриваемой системе одинаковые требования будут предъявляться ко всем серверам обработки информации центрального уровня.

На основании модели угроз и описания объекта осуществляется автоматизированная классификация автоматизированной системы АС (скрипт classification):

– определяется класс ИСПДн (в соответствии с руководящими документами ФСТЭК России);
– тип нарушителя (путем сравнения с описанием типов нарушителя в базовой модели);
– класс в соответствии с РД, результат классификации записывается в текстовый файл и включается отдельной диаграммой в модель.

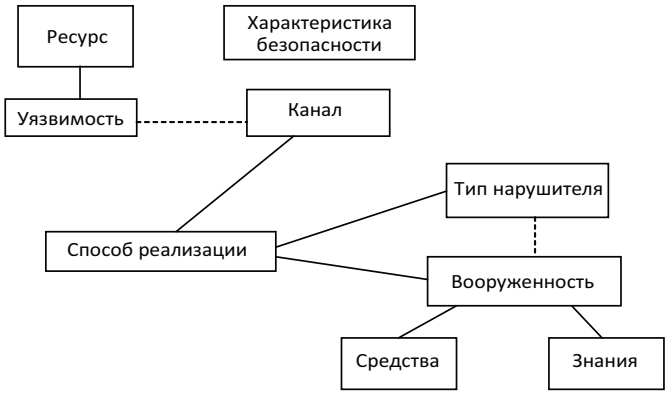


Рис. 3. Связь понятий, используемых при разработке детализированной модели угроз

Проверка модели угроз

Проверка модели осуществляется на основании формальных правил. Построенная детализированная модель угроз анализируется, в частности, по следующим параметрам:

- правильность классификации нарушителя (осуществляется сравнением классов, описывающих возможности нарушителя данного объекта, с классами базовой модели, при этом проверяется наличие всех исходных данных);
- соответствие детализированных угроз угрозам верхнего уровня, то есть для каждой угрозы верхнего уровня должен существовать, как минимум, один способ реализации;
- для каждого канала реализации должен быть, как минимум, один способ реализации;
- для систем, имеющих подключения к каналам общего пользования, должны быть предусмотрены угрозы удаленного доступа (для всех объектов, имеющих такое подключение);
- для информации, передающейся по открытым каналам, должны рассматриваться угрозы перехвата, модификации (в зависимости от характеристик безопасности);
- для ИСПДн 1 и 2 класса, ИС, использующих СКЗИ, должны быть учтены угрозы утечки по техническим каналам.

Разработка информационной системы в защищенном исполнении

При построении модели защиты необходимо учитывать модель объекта; модель угроз безопасности; требования регуляторов к функциональным возможностям системы защиты.

При формировании модели системы защиты информационной системы для каждого технического средства формируется перечень актуальных угроз. Данный перечень формируется путем группировки сходных угроз безопасности ресурсов, размещенных на данном сервере, для некоторых типов угроз прослеживаются ассоциации с формами представления информации. В результате образуется перечень угроз вида «утечка информации, обрабатываемой на сервере А, по техническим каналам», «внедрение на сервер А вредоносных программ» и т. п. Для угроз, осуществляемых по сети, проводится проверка стереотипов каналов связи для определения «точек» установки средств межсетевого экранирования, обнаружения атак и других средств защиты, обеспечивающих безопасное подключение к открытым сетям.

На основании данного перечня автоматически осуществляется предварительный выбор требуемых механизмов защиты, который затем дорабатывается экспертным путем.

На основании модели СЗИ и модели объекта генерируется модель информационной системы в защищенном исполнении. В рамках данного подхода в качестве PIM рассматривается бизнес-модель распределенной системы, которая затем отображается на «технологическую платформу» – систему защиты информации[9].

Результатом данного отображения будем PSM – модель системы в защищенном исполнении, по которой в дальнейшем возможна генерация кода. Для реализации данного отображения в соответствии с технологией MDA также необходимо разработать метамодель преобразования и инструмент, его реализующий.

Этот механизм позволяет сравнительно легко модернизировать систему защиты информации, сохраняя согласованность компонентов на уровне модели и, следовательно, на уровне кода. При использовании данной технологии достаточно внести изменение только в модель СЗИ, а затем повторить процедуру преобразования, фактически отобразить бизнес-модель на другую технологическую платформу.

Требования к системам защиты информации формулируются не только заказчиком, но и государственными регуляторами в сфере информационной безопасности, для многих систем требуется аттестация по требованиям безопасности информации и сертификация применяемых средств защиты информации, что делает нецелесообразным разработку собственных механизмов защиты для каждого бизнес-приложения, так как это влечет дополнительные затраты и сложности, иными словами, целесообразно использование готовых средств защиты информации.

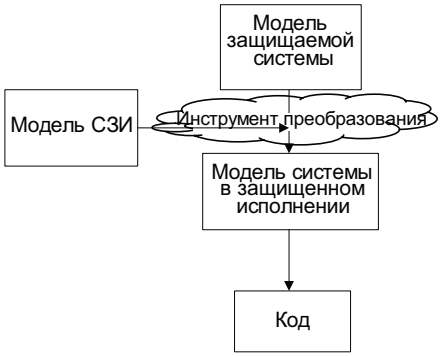


Рис. 4. Создание системы в защищенном исполнении

При этом необходимо обеспечить комплексность защиты и достаточность применяемых механизмов.

В рамках данного подхода для каждого ресурса на основании модели защиты автоматически определяется перечень подключаемых механизмов защиты, который заносится в свойства класса в инструмент Security. Применительно к каждому из основных механизмов защиты инструмент преобразования формирует интерфейсы защиты, определяющие требуемый класс СВТ, а также перечень настроек. Например, с точки зрения системы разграничения доступа на основе сценариев и диаграмм действия модели объекта формируется матрица доступа, в случае необходимости определяется уровень допуска субъектов и уровень конфиденциальности объектов. Полученная информация выводится в текстовом виде и может быть использована как для автоматической настройки средств защиты, так и администратором безопасности.

Полученная модель системы в защищенном исполнении проверяется на соответствие требованиям регуляторов. Для каждого требования регулятора автоматически указывается правило модели защиты, его реализующее, и предлагаемая настройка средств защиты.

Вывод

Рассматриваемая система поддержки разработки автоматизированных систем в защищенном исполнении не может в автоматическом режиме решать нестандартные задачи или предлагать нетривиальные решения, однако гибкость предложенного инструмента позволяет вносить любые изменения в построение системы защиты либо самого объекта, исключать и добавлять угрозы на всех этапах работ, при этом инструмент обеспечит актуальность всех представлений системы, включая документацию. Использованный в инструменте метод моделирования позволит разработчикам СЗИ легче ориентироваться в современных сложных распределенных ИС, находить их уязвимые места и предлагать нестандартные решения.

Примечания

[1] См.: *Ефремов А.* Понятие и виды конфиденциальной информации [Электронный ресурс] // Сайт В.Б. Наумова. [М., 2009]. URL: http://www.russianlaw.net/law/doc/a90.htm (дата обращения: 4.11.2009).

2 В том числе и благодаря ФЗ «О персональных данных», требующему привести информационные системы персональных данных в соответствие с законодательством до 1 января 2010 г.

3 См.: *Jurjens J.* Secure Systems Development with UML, Springer Academic Publishers, 2004 [Электронный ресурс] // Сайт программы Emule. [М., 2009]. URL: ed2k://|file|Secure_Systems_Development_with_UML_(Springer-2005).pdf|2459327|7B6A7A0EA87A02B AA419A69D9EC69A0E|/ (дата обращения: 4.11.2009); *Lodderstedt T., Basin D., Doser J.* SecureUML: A UML-Based Modeling Language for Model-Driven Security [Электронный ресурс]. [М., 2009]. URL: http://kisogawa.inf.ethz.ch/WebBIB/publications-softech/papers/2002/0_secuml_uml2002.pdf (дата обращения: 4.11.2009); *Peterson M.J., Bowles J.B., Eastman C.M.* UMLpac: An Approach for Integrating Security into UML Class Design [Электронный ресурс]. [М., 2009]. URL: http://www.cse.sc.edu/~yoncek/REU/PetersonPaper.pdf (дата обращения: 4.11.2009).

4 См.: *Lodderstedt T., Basin D., Doser J.* Model Driven Security from UML Models to Access Control Infrastructures [Электронный ресурс]. [М., 2009]. URL: http://www.inf.ethz.ch/personal/basin/pubs/mdac-tosem.pdf (дата обращения: 4.11.2009); *Basin D., Doser J.* Model Driven Security for Process-Oriented Systems [Электронный ресурс]. [М., 2009]. URL: http://kisogawa.inf.ethz.ch/WebBIB/publications/papers/2003/ p344- odderstedt.pdf (дата обращения: 4.11.2009).

5 Для систем, отнесенных к компетенции ФСБ России, и в случае применения криптографических (шифровальных) средств.

6 Характеристика безопасности моделируется классом. При необходимости внедрения дополнительной характеристики безопасности создается класс, который будет обработан инструментом преобразования.

7 См. в наст. номере: *Приезжая А.Н.* Анализ нормативно-методических документов в области защиты персональных данных.

8 Данные модели построены на основании практических разработок автора. Модели угроз и потенциального нарушителя, положенные в основу базовой модели, прошли согласование во ФСТЭК России и ФСБ России.

9 *Приезжая А.Н.* Технологии встраивания функций безопасности в бизнес процессы // Вестник РГГУ. 2009. № 10. Сер. «Информатика. Защита информации. Математика». С. 71–84.

Ю.К. Сергеев

АНАЛИЗ НЕКОТОРЫХ МЕХАНИЗМОВ УПРАВЛЕНИЯ ПАМЯТЬЮ ВИРТУАЛЬНЫХ МАШИН

Эволюция вредоносного ПО, выражающаяся в росте возможностей данных программ по осуществлению несанкционированных действий в ИС, ведет за собой развитие средств защиты. С каждым годом вирусы, сетевые черви и троянские программы становятся все интеллектуальнее, так что для борьбы с ними применяются все новые и новые средства, ищутся новые подходы. Зная установленные средства защиты информации (СЗИ) на атакуемом хосте, злоумышленник всегда может разработать новую технологию для скрытия вредоносного кода от заданных средств по борьбе с вредоносным ПО. Автор ставит целью спроектировать такую систему, в которой существует возможность обеспечить невидимость СЗИ для злоумышленника и/или вредоносных программ в рамках той ОС, в которую осуществляется вторжение, путем вынесения СЗИ за ее рамки. Для реализации такого рода архитектуры предлагается опираться на технологии виртуализации с применением гипервизора. Ввиду открытости исходных кодов (Open Source) для достижения этой цели автором используется гипервизор Xen. Для того чтобы исследовать возможность применения данной технологии, необходимо рассмотреть механизмы, позволяющие изолировать виртуальные машины друг от друга. Одним из ключевых механизмов является управление оперативной памятью виртуальных машин.

Ключевые слова: гипервизор, механизмы управления памятью, средства защиты информации, изоляция виртуальных машин.

Управление памятью в системах, использующих виртуализацию, является одним из важнейших процессов с точки зрения безопасности. Неверная архитектура или недостаточность проверок выполнения тех или иных операций с памятью может приводить к серьезным последствиям – реализации угрозы НСД памяти

ОС в обход любых механизмов защиты, функционирующих внутри данной системы.

Выделение физической памяти виртуальным машинам должно осуществляться под контролем гипервизора с помощью встроенных механизмов проверки валидности таких операций.

Обеспечение безопасности при выделении и освобождении памяти

Для архитектуры x86 характерно выделение и освобождение оперативной памяти постранично. При создании нового домена, память которого обычно варьируется от 2 до 8 Гб, невозможно гарантировать, что найдется непрерывное пространство памяти такой длины, следовательно, физическая память обычно фрагментирована. Для создания имитации непрерывности пространства физической памяти для виртуальных машин (доменов) гипервизор, функционирующий в нулевом кольце процессора, реализует механизм объединения свободных блоков физической памяти в области псевдофизической памяти, т. е. создается уровень абстракции для доменов при работе с физической памятью.

Xen ведет таблицу соответствия физических (Machine Address, MA) и псевдофизических (Pseudo-Physical Address, PPA) адресов: MA->PPA. Для обратного преобразования к каждому домену прикрепляется таблица, содержащая обратное соответствие: PPA->MA.

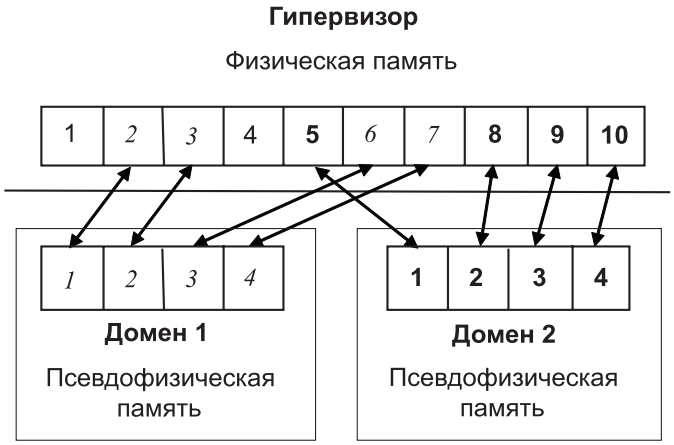


Рис. 1. Сопоставление физической и псевдофизической памяти

Возвращаясь к выделению памяти виртуальным доменам, стоит отметить, что каждый домен имеет максимальное и текущее выделение физической памяти, т. е. некоторые области памяти могут периодически выделяться разным доменам. Таким образом, в случае интенсивного использования оперативной памяти могут возникать случаи, когда та или иная область физической памяти после записи в нее информации может быть освобождена доменом и быть доступна другому. Если приложение, функционирующее в рамках виртуального домена, не осуществляет очистку памяти при ее освобождении и ОС также не контролирует данный процесс, может возникнуть угроза утечки информации по памяти.

Проведем эксперимент. На исходном компьютере под управлением Xen с 1024 Мб оперативной памяти зафиксируем максимальное количество памяти для привилегированного домена:

```
root@sonic:/etc/xen# xm mem-set 0 400
root@sonic:/etc/xen# xm list
Name ID Mem VCPUs State Time(s)
Domain-0 0 400 2 r----- 764.3
```

Запустим гостевой домен winxp1 (256 Mb) под управлением Windows XP:

```
root@sonic:/etc/xen# xm create winxp1
Using config file "./winxp1".
Started domain winxp1
root@sonic:/etc/xen# xm list
Name ID Mem VCPUs State Time(s)
Domain-0 0 400 2 r----- 798.1
winxp1 8 256 1 -b---- 51.7
```

Сразу после загрузки и получения доступа к рабочему столу сделаем полный дамп памяти домена winxp1 с помощью специально разработанной программы XenDumpHVM[1]:

```
root@sonic:/etc/xen# ./XenDumpHVM 8 1 > winxp1.core
```

Данная копия памяти будет использована в качестве эталона системы сразу после загрузки. Затем запустим программу «Сапер» в данном домене и также сделаем дамп памяти, а затем завершим его работу:

```
root@sonic:/etc/xen# ./XenDumpHVM 8 1 > winxp1.core2
root@sonic:/etc/xen# xm shutdown winxp1
```

Запустим другой домен winxp2 (512 Mb) также под управлением Windows XP для того, чтобы проверить путем создания третьего дампа памяти, что информация о запущенной в другой ОС программе будет доступна в оперативной памяти созданного:

```
root@sonic:~/.workspace/XenDumpHVM/Debug# xm create winxp2
```

```
Using config file "/etc/xen/winxp2".
Started domain winxp2
root@sonic:~/.workspace/XenDumpHVM/Debug# xm list
Name ID Mem VCPUs State Time(s)
Domain-0 0 400 2 r----- 1012.3
winxp2 9 512 1 r----- 5.6
root@sonic:/etc/xen# ./XenDumpHVM 9 1 > winxp2.core
kirion@susic:~$ ls -lh winxp*
-rw-r--r-- 1 kirion kirion 267M 2009-10-26 23:40 winxp1.core
-rw-r--r-- 1 kirion kirion 267M 2009-10-26 23:40 winxp1.core2
-rwx------ 1 kirion kirion 525M 2009-10-26 23:41 winxp2.core
```

В результате сравнения двух первых файлов видно, что в странице памяти № 57159 появились новые данные, соответствующие слепку памяти программы «Сапер», а большая часть оперативной памяти 3-го кольца процессора содержит нули, так как компьютер был только недавно запущен.

```
00000000 73 74 61 72 74 20 2D 20 67 6D 66 6E 20 3D 2C 30 0A 53  start - gmfn = 0.s
00000012 FF 00 F0 53 FF 00 F0  53 FF 00 F0 53 FF  00  F0 53 FF 00  ...s...s...s...s..
00000024 F0 53 FF 00 F0  53 FF 00 F0 53 FF  00  F0 A5 FE 00 F0  87  .s...s...s........
00000036 E9 00 F0  53 FF 00 F0  53 FF 00 F0 53 FF  00  F0 53 FF 00  ...s...s...s...s..
00000048 F0 57 EF 00                                                .s...n...M...A
0000005a F8 00 F0 FF                                                ...9...Y.......
0000006c F0 D2 EF 00                                                ........n...s
0000007e FF 00  F0 4A                                               ...s.......>..
00000090 C0 53 FF 00                                                .s...s...s...s
000000a2 FF 00 F0  53                                               ...s...s...s..
000000b4 F0  53 FF 00 F0  53 FF 00 F0  53 FF 00 F0  53 FF 00 F0  53  s...s...s...s
000000c6 FF 00 F0  53  FF 00 F0  53 FF 00 F0  53 FF 00 F0  53 FF 00  ...s...s...s...s..
000000d8 F0  53 FF 00 F0  53 FF 00 F0  53 FF 00 F0  53 FF 00 F0 53  .s...s...s...s..
000000ea FF 00 F0  53 FF 00 F0  53 FF 00 F0  53 FF 00 F0  53 FF 00  ...s...s...s...s..
```

The pattern you requested was not found.

End of file reached.

```
0e16fc1e  66 85 C0 75 E5 FF 75 E0 FF 75 08 57 FF D6 83 C4 0C 85   f..u..u..u.w.....
0e16fc30  C0 75 BD 8D 4D D0 E8 E5 BE 65 6E 64 20 2D 20 67 6D 66  .u..M....end - gmf
0e16fc42  6E 20 3D 20 35 37 31 35 38 0A 73 74 61 72 74 20 2D 20   n = 57158.start -
0e16fc54  67 6D 66 6E 20 3D 20 35 37 31 35 39 0A 40 00 00 00 84  gmfn = 57159.@....
0e16fc66  6A E4 73 00 00 00 00 2E 50 41 56 43 52 65 73 6F 75 72  j.s.....PAVCResour
0e16fc78  63 65 45 78 63 65 70 74 69 6F 6E 40 40 00 00 00 00 84  ceException@@.....
0e16fc8a  6A E4 73 00 00 00 00 2E 3F 41 56 43 52 65 73 6F 75 72  j.s.....?AVCResour
0e16fc9c  63 65 45 78 63 65 70 74 69 6F 6E 40 40 00 00 00 00 84  ceException@@.....
0e16fcae  6A E4 73 00 00 00 00 2E 3F 41 56 43 55 73 65 72 45 78  j.s.....?AVCUserEx
0e16fcc0  63 65 70 74 69 6F 6E 40 40 00 00 00 00 84 6A E4 73 00  ception@@.....j.s.
0e16fcd2  00 00 00 2E 3F 41 56 43 43 6C 69 65 6E 74 44 43 40 40  ....?AVCClientDC@@
0e16fce4  00 84 6A E4 73 00 00 00 00 2E 3F 41 56 43 57 69 6E 64  ..j.s.....?AVCWind
0e16fcf6  6F 77 44 43 40 40 00 84 6A E4 73 00 00 00 00 2E 3F 41  owDC@@..j.s.....?A
0e16fd08  56 43 50 61 69 6E 74 44 43 40 40 00 00 84 6A E4 73 00  VCPaintDC@@...j.s.
```

Рис. 2. Сравнение дампов памяти доменов путем поиска паттерна

В качестве паттерна для поиска было выбрано слово «AVCResourceException». В первом файле такое слово не встречается. В третьем файле (дампе памяти домена winxp2) те же самые данные с использованием паттерна «AVCResourceException» доступны, но по другому адресу, при этом информация после искомого слова не исказилась.

0650fd3c	40 00 00 00 00 84 6A E4 73 00 00 00 00 2E 3F 41 56 43	@.....j.s.....?AVC
0650fd4e	43 6F 6D 62 6F 42 6F 78 45 78 40 65 6E 64 20 2D 20 67	ComboBoxEx@end– g
0650fd60	6D 66 6E 20 3D 20 32 35 36 32 36 0A 73 74 61 72 74 20	mfn = 25626.start
0650fd72	2D 20 67 6D 66 6E 20 3D 20 32 35 36 32 37 0A 40 00 00	– gmfn = 25627.@..
0650fd84	00 84 6A E4 73 00 00 00 00 2E 50 **41 56 43 52 65 73 6F**	..j.s.....P**AVCReso**
0650fd96	**75 72 63 65 45 78 63 65 70 74 69 6F 6E** 40 40 00 00 00	**urceException**@@...
0650fda8	00 84 6A E4 73 00 00 00 00 2E 3F **41 56 43 52 65 73 6F**	..j.s.....?**AVCReso**
0650fdba	**75 72 63 65 45 78 63 65 70 74 69 6F 6E** 40 40 00 00 00	**urceException**@@...
0650fdcc	00 84 6A E4 73 00 00 00 00 2E 3F 41 56 43 55 73 65 72	..j.s.....?AVCUser
0650fdde	45 78 63 65 70 74 69 6F 6E 40 40 00 00 00 00 84 6A E4	Exception@@.....j.
0650fdf0	73 00 00 00 00 2E 3F 41 56 43 43 6C 69 65 6E 74 44 43	s.....?AVCClientDC
0650fe02	40 40 00 84 6A E4 73 00 00 00 00 2E 3F 41 56 43 57 69	@@..j.s.....?AVCWi
0650fe14	6E 64 6F 77 44 43 40 40 00 84 6A E4 73 00 00 00 00 2E	ndowDC@@..j.s.....
0650fe26	3F 41 56 43 50 61 69 6E 74 44 43 40 40 00 00 84 6A E4	?AVCPaintDC@@...j.

Рис. 3. Выделение памяти доменом
без ее предварительного освобождения

Таким образом, показан пример реализации утечки информации по памяти, возможность реализации которой происходит из-за отсутствия механизмов очистки памяти перед ее выделением другому домену.

Для предотвращения такого рода утечек необходимо реализовать такую функцию для страниц, подключаемых к домену. Операция присваивания страниц домену реализована в файле page_alloc.c[2] в функции alloc_domheap_pages(), которая вызывает универсальную функцию выделения памяти alloc_heap_pages(). Последняя функция получает адрес, с которого она должна начать поиск, и количество страниц, которые нужно выделить. Затем осуществляет поиск после указанного адреса ближайших свободных страниц и возвращает информацию о них вызвавшей ее функции. Очистку можно производить непосредственно перед тем, как отдать страницы на присвоение домену.

Механизмы разделения памяти между доменами

Основным механизмом разделения памяти между доменами являются таблицы доступа (Grant tables). За каждым доменом закреплена собственная таблица, представляющая собой структуру данных, с помощью которой гипервизор определяет, какие привилегии были даны тому или иному домену для доступа к данному. Записи в таблице называются ссылками доступа (grant references). Создание и удаление ссылок осуществляется путем прямого доступа к таблице, что позволяет изменять привилегии доступа других доменов к областям памяти исходного без привлечения к этому процессу гипервизора. Для модификации таблицы доступа домена могут использоваться следующие четыре операции:

- предоставление права доступа субъекту (Grant foreign access). Создает новую запись в таблице и заполняет указанными правами доступа субъекта. Привилегии проверяются гипервизором каждый раз, когда какой-либо домен запрашивает с помощью гипервызова (hypercall) ссылку для подключения указанной области памяти в свое пространство памяти (процедура маппирования – mapping);
- изъятие права доступа у субъекта (End foreign access). Если ссылка доступа в данный момент не используется никаким доменом, то данная операция удаляет запись доступа к заданной области;
- предоставление права передачи данных субъекту (Grant foreign transfer). Создает новую запись в таблице и заполняет указанными правами доступа субъекта. Привилегии проверяются гипервизором каждый раз, когда какой-либо домен запрашивает с помощью гипервызова передать подключенную ранее область памяти обратно в исходный домен;
- изъятие права передачи данных субъекту (End foreign transfer).

Таким образом, гипервизор реализует дискреционный принцип разграничения доступа, где:

- субъекты доступа – домены;
- привилегированные субъекты доступа – привилегированные домены;
- объекты доступа – любые области памяти любых доменов;
- монитор обращений – гипервизор Xen;
- правила разграничения доступа – таблица доступа для каждого домена с указанием привилегий других доменов при доступе к данному.

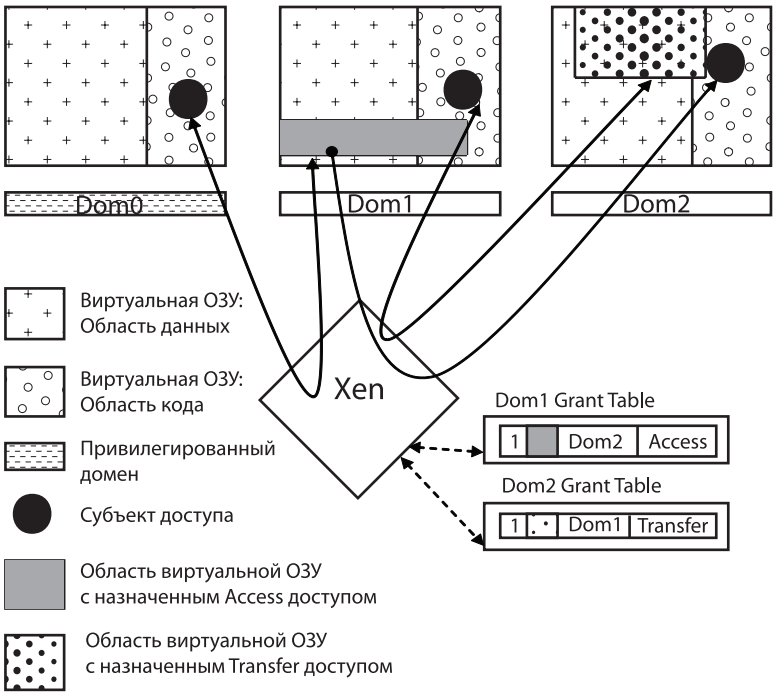


Рис. 4. Механизм дискреционного контроля доступа к памяти домена

При данном принципе разграничения доступа существует вероятность организации сценария «Троянский конь». Приведем пример. Домен A предоставляет доступ к области памяти {am-an} домену B. Домен B предоставляет доступ к области памяти {bk-bl} домену C. Для получения информации из домена A злоумышленнику достаточно установить программный агент в домене B для осуществления копирования данных {am-an} из A в область {bk-bl}, доступную C. Другими словами, возможно реализовать канал передачи информации между виртуальными компьютерами без использования сетевых технологий. Интересным является также тот факт, что для реализации этого канала нет необходимости встраивать какой-либо скрытый агент на атакуемый компьютер. Таким образом, данная атака будет «невидима» для любых существующих средств защиты, функционирующих внутри домена A, а ее обнаружение будет возможно только из привилегированных доменов.

Сегодня существует специальный модуль XSM (Xen Security Module), позволяющий интегрировать с гипервизором сторонние механизмы принудительного контроля доступа. В качестве архитектуры (framework) для реализации такого рода политик может быть использована Flask[3]. На основе этой архитектуры возможно формирование частных политик принудительного контроля доступа в зависимости от поставленных задач.

Реализация Flask относительно громоздкая, но зато очень гибкая, так как позволяет написать правила разграничения доступа для большинства политик принудительного контроля доступа. В случае определения необходимости использования только одной политики достаточно использовать API модуля XSM для написания собственной реализации.

Для обеспечения «невидимости» средства защиты, размещенного в выделенном домене безопасности, из защищаемого домена необходимо использовать политику принудительного доступа, которая не будет позволять «читать вверх» и «писать вверх», т. е. реализовать в дополнение к существующей дискреционной политике разграничения доступа к памяти между доменами следующую:

субъекты отношений (домены): A, B;

уровни доступа: System (S), Guard (G), Common (C);

Типы доступов: R – чтение памяти, W – запись в память;

C = {S,G,C} – решетка ценности;

если C(A) > C(B), то A $\xleftarrow{R}$ B и A $\xrightarrow{W}$ B;

если C(A) = C (B), то A $\xleftarrow{R}$ B, B $\xleftarrow{R}$ A и A $\xrightarrow{W}$ B, B $\xrightarrow{W}$ A;

если C(A) < C(B), то A запрещен любой доступ к B.

Иллюстрация данной политики приведена на рис. 5:

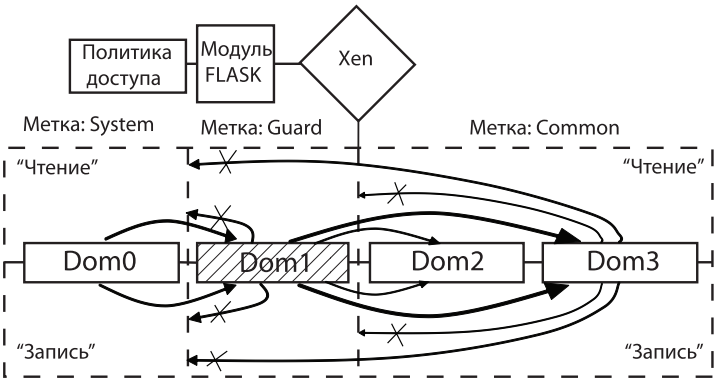


Рис. 5. Политика доступа для обеспечения «невидимости»

При этом изменять уровни доступа домена должен только пользователь, имеющий административные привилегии в домене с уровнем доступа System.

Заключение

В проведенном исследовании были:
- рассмотрены несколько основных механизмов управления памятью, реализованных в гипервизоре Xen;
- найдена уязвимость, связанная с выделением памяти домену после ее освобождения другим доменом;
- предложен путь нейтрализации данной уязвимости путем встраивания операций по очистке памяти в функцию alloc_heap_pages();
- рассмотрены механизмы разграничения доступа памяти для общей работы доменов с одной и той же областью памяти;
- предложена для реализации политика, позволяющая обеспечить «невидимость» средств защиты по оперативной памяти для любых вредоносных программ, функционирующих в рамках защищаемой ОС.

На основе полученных результатов можно сделать вывод о том, что построение архитектуры безопасности, в которой средство защиты становится «невидимым» для вредоносной программы вне зависимости от его возможностей в рамках защищаемой ОС, осуществимо с применением гипервизора Xen для рассмотренных операций с памятью.

В последующем исследования могут быть направлены на другие взаимодействия между доменами:
- механизмы обмена сообщениями между доменами, реализуемые через оперативную память;
- механизмы взаимодействия через сетевую подсистему Xen;
- возможности несанкционированного подключения блочных устройств хранения данных других доменов.

Примечания

[1] *Сергеев Ю.К.* Использование технологий виртуализации для защиты информации // Вестник РГГУ. 2009. № 10. Сер. «Информатика. Защита информации. Математика». С. 98–109.

2 См.: *Spector S.* Xen Source Code Overview [Электронный ресурс] // Сайт Xen. [М., 2009]. URL: http://blog.xen.org/index.php/2009/06/15/xen-34-source-code-overview/ (дата обращения: 4.11.2009).

3 *Spencer R., Smalley S., Loscocco P., Hibler M., Andersen D., Lepreau J.* The Flask Security Architecture: System Support for Diverse Security Policies // Proceedings of the Eighth USENIX Security Symposium. 1999. Aug. P. 123–139. См. также: *Coker G.* Xen Security Modules (XSM). National Information Assurance Research Lab, National Security Agency (NSA): Presentation, 17 April 2007.

М.В. Левыкин

АНАЛИЗ ЗАЩИЩЕННОСТИ ШТАТНОГО МЕХАНИЗМА КОНТРОЛЯ ДОСТУПА К РЕЕСТРУ В ЯДРЕ ОС WINDOWS XP*

Системный реестр играет ключевую роль в конфигурировании и управлении ОС Windows. Это хранилище общесистемных и пользовательских параметров. Реестр не является статической совокупностью хранящихся на жестком диске данных. Он представляет собой набор различных структур, которые хранятся в памяти компьютера и поддерживаются ядром и исполнительной системой. Реестр используется: в ходе загрузки системы, при загрузке Explorer и других компонентов ОС, при установке приложений и драйверов, при запуске приложений и драйверов и т. д.

В настоящее время трудно недооценить роль системного реестра в вопросах, связанных с безопасностью ОС Windows, так как функционирование практически всех современных средств скрытия так или иначе связано с системным реестром.

Ключевые слова: системный реестр, механизм обратного вызова реестра, исполнительная система, диспетчер системных сервисов, диспетчер конфигурации, ядро ОС Windows, драйвер.

В данной статье будет рассмотрен весь путь доступа к системному реестру от пользовательского режима до *диспетчера конфигурации*, компонента исполнительной системы режима ядра. Далее будет описан предложенный разработчиками ОС механизм, позволяющий контролировать доступ к системному реестру на уровне исполнительной системы, приведены достоинства этого механизма. Будет предложен способ обхода штатного механизма контроля доступа к реестру. В заключение будут представлены полученные результаты и выводы работы.

* Работа выполнена при поддержке РФФИ, грант № 07-07-00236.

Цели и задачи

Цель данной работы провести анализ защищенности штатного механизма контроля доступа к реестру в ядра ОС Windows XP с целью выявления его уязвимостей.

Для достижения поставленной цели необходимо решить следующие задачи:

- описать систему доступа к реестру с уровня приложения до уровня ядра;
- описать работу штатного механизма контроля доступа к реестру в ярде ОС Windows XP;
- обосновать возможность обхода штатного механизма контроля доступа к реестру в ярде ОС Windows XP.

Анализ использованных источников и литературы

При написании работы была использована литература по двум основным тематикам: архитектуре ядра ОС Windows и программированию в ядре[1]. Необходимо заметить, что документации по архитектуре Windows достаточно много. Она хорошо представлена в среде для разработчиков драйверов DDK (Driver Development Kit) в книгах Марка Руссиновича и Дэвида Соломона. Однако вследствие того что ОС Windows XP является коммерческим программным продуктом, многие механизмы ядра скрыты от пользователей этой системы. В связи с этим довольно часто для воссоздания системных механизмов приходится прибегать к отладке уровня ядра, которую можно осуществлять с помощью удобного программного средства WinDbg.

Программирование в ядре ОС Windows также описывается в DDK. Одними из лучших источников являются книги В. Солдатова[2]. Безусловно, самым полным справочником по программированию является сайт разработчиков компании Microsoft – msdn.com.

Системный реестр

Проследим на рис. 1 этапы работы с реестром с прикладного уровня до уровня ядра. В качестве приложения редактирования реестра в общем случае может выступать любое пользовательское приложение. Однако на практике наиболее часто этим приложением является Registry Editor – стандартный *редактор реестра*.

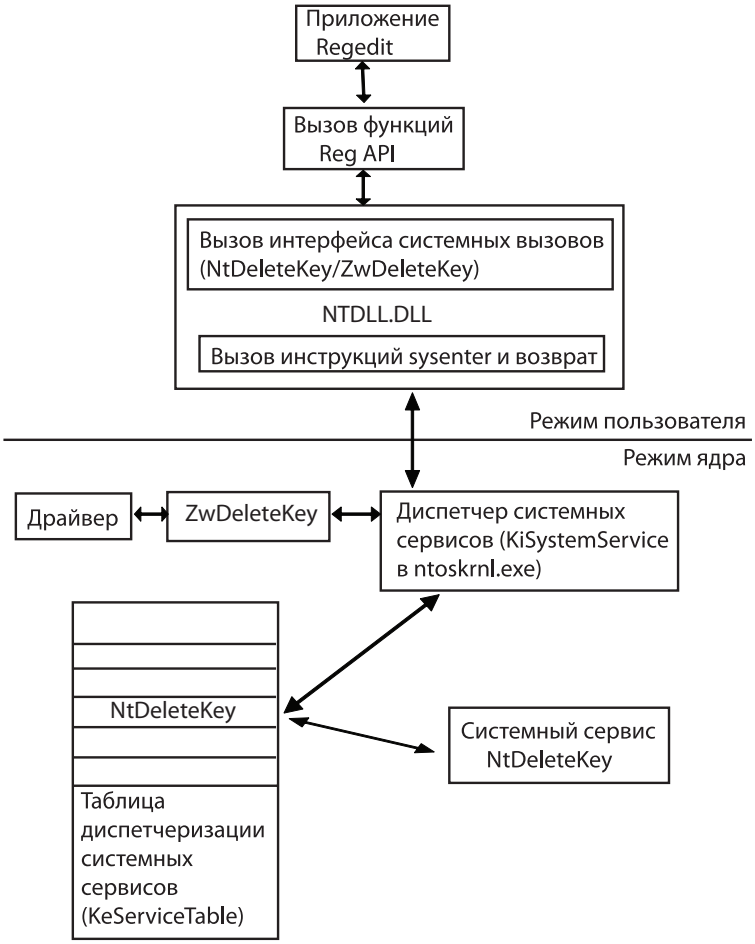


Рис. 1. Системный сервис удаления ключа реестра уровня ядра

Этот редактор, как и любое другое приложение, использует интерфейс системных вызовов для работы с реестром. Назовем этот набор *Reg API*.

Все эти функции – функции-обертки, которые вызывают непосредственно функции интерфейса системных вызовов (функции с префиксом Nt/Zw...). Системная библиотека ntdll.dll экспортирует API-функции работы с реестром в библиотеку advapi32.dll, которая доступна из пользовательского режима (рис. 2).

```
lkd> dd ntdll!zwcreateprocess
7c90d130 00002fb8 0300ba00 12ff7ffe 900020c2
lkd> dd ntdll!ntcreateprocess
7c90d130 00002fb8 0300ba00 12ff7ffe 900020c2
```

Рис. 2. Экспорт функции открытия ключа реестра из ntdll

Вызов функции из ntdll приводит к тому, что происходит программное прерывание – вызывается выполнение инструкции процессора sysenter (на Intel x86 архитектуре), которая переводит контекст выполнения в режим ядра и вызывает *диспетчер системных сервисов* KeSystemService.

Диспетчер системных сервисов проверяет возможность доступа к данным пользовательского режима, передаваемым в качестве аргументов для выполнения *системного сервиса*. Затем, найдя нужный адрес системного сервиса (в нашем примере NtDeleteKey) в таблице *диспетчеризации системных сервисов* KeServiceTable, вызывает его выполнение.

Из рис. 1 видно, что API-функции работы с реестром доступны как на уровне пользователя, так и на уровне ядра. Это связано с тем, что реестр используется и драйверами, и приложениями.

Однако функции ZwXXXKey работы с реестром уровня ядра экспортируются ядром ntoskrnl.exe, а функции NtXXXKey – нет, потому что разработчики ОС идеологически определили единую точку входа, с помощью которой осуществляется вызов системных сервисов – диспетчер системных сервисов.

Именно поэтому диспетчер системных сервисов, а точнее таблица диспетчеризации системных сервисов, является целью большинства антивирусных средств и средств сокрытия.

На рис. 3 представлен пример перехвата антивирусом Касперского системного сервиса удаления ключа реестра.

Как видно из рис. 3, при вызове системного сервиса NtDeleteKey либо на уровне ядра (драйвером), либо на уровне приложения (прикладной программой) будет вызван не оригинальный системный сервис NtDeleteKey, а функция Касперского (условно назовем ее KAVNtDeleteKey), которая, произведя свою собственно проверку легальности данного вызова системного сервиса, примет решение о вызове его или нет.

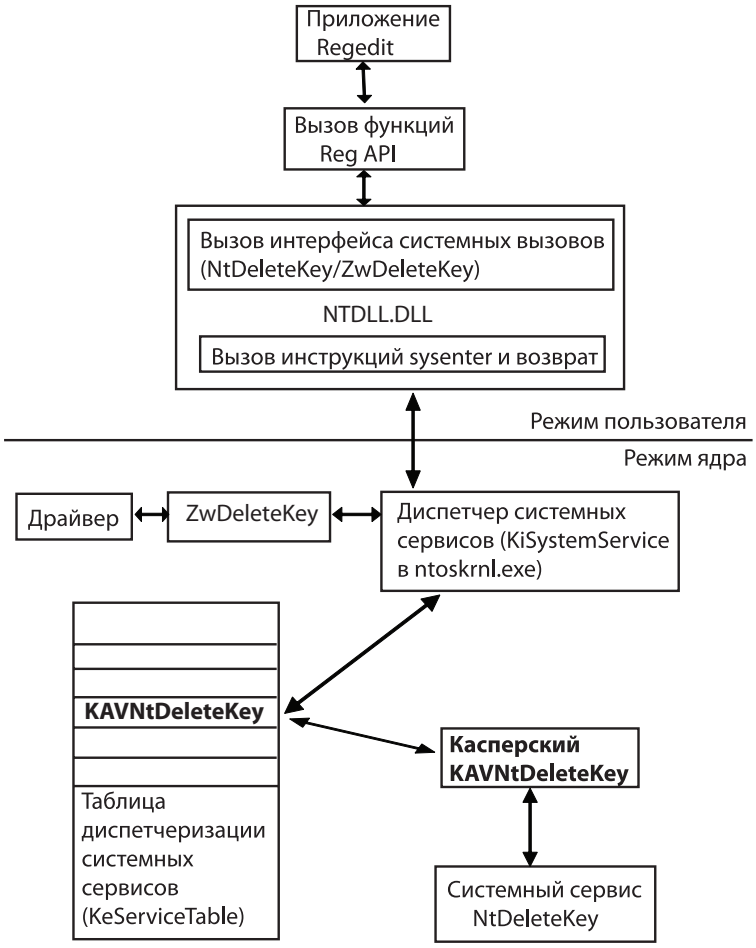


Рис. 3. Перехват антивирусом Касперского системного сервиса удаления ключа реестра

Соответственно, контролируя все системные сервисы работы с реестром, можно говорить о контроле доступа к нему.

Именно поэтому разработчики ОС предложили новый штатный механизм контроля доступа к реестру на более низком уровне системы.

Рассмотрим схему, представленную на рис. 4. На ней изображен в общем виде механизм контроля доступа к реестру в ядре ОС.

Этот механизм обратного вызова реестра[3] впервые появился в ОС Windows XP и поддерживается в более поздних вер-

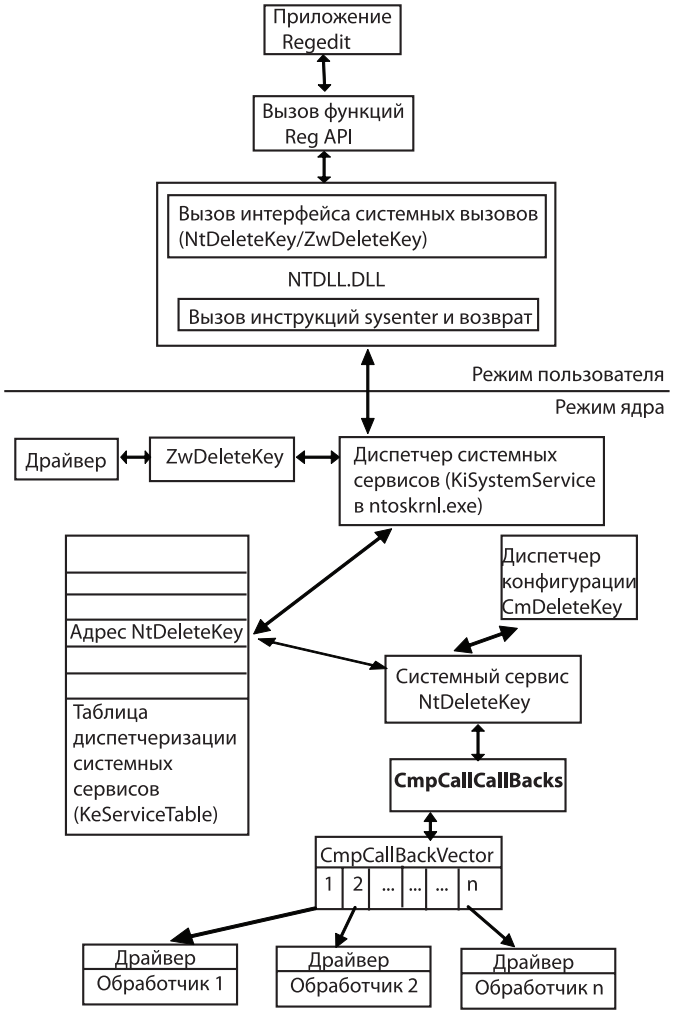


Рис. 4. Механизм обратного вызова реестра
на примере системного сервиса NtDeleteKey

сиях ОС Windows. Согласно этому механизму драйвер регистрирует в диспетчере конфигурации функцию обратного вызова – обработчик этого системного сервиса реестра. Далее при вызове системного сервиса вызывается выполнение обработчика, а в качестве входящих данных ему передаются структуры реестра до их изменения и после. То есть функция обратного вызова при выполнении системного сервиса получает в виде структуры данных (поле данных IRP пакета) информацию до внесения ее в реестр на уровне диспетчера конфигурации и после этого внесения. Следовательно, средство обнаружения может контролировать процесс доступа к реестру на уровне более низком, чем при перехвате системных сервисов.

Согласно документации Windows, предоставляемой для разработчиков, механизм обратного вызова служит для блокирования, фильтрации и аудита доступа к реестру[4].

Однако политика компании Microsoft предполагает, что работа данного механизма скрыта от разработчиков ПО. Разработчику антивирусного или антируткитового ПО предлагается только функция, позволяющая установить или удалить свой собственный обработчик – CmRegisterCallback. При этом не предоставляется возможность просматривать все установленные обработчики.

В случае удаления обработчика другими драйверами, также не будет получен сигнал об удалении. Известно только максимальное количество обработчиков, которое не может быть больше 8-ми. Другими словами предполагается, что обработчики независимы друг от друга.

Главным достоинством этого механизма является то, что он действует в ядре ОС на уровне диспетчера конфигурации. Следовательно, как уже говорилось, средство обнаружения или сокрытия, построенное с помощью этого механизма, находится на более низком уровне, чем перехват системных сервисов, что удовлетворяет условию невлияния[5] (рис. 5).

Однако закрытость данного механизма (в частности, отсутствие обратной связи) и, как следствие, невозможность его полного контроля (так как могут быть другие установленные в системе обработчики, о которых вообще ничего неизвестно), является главным его недостатком.

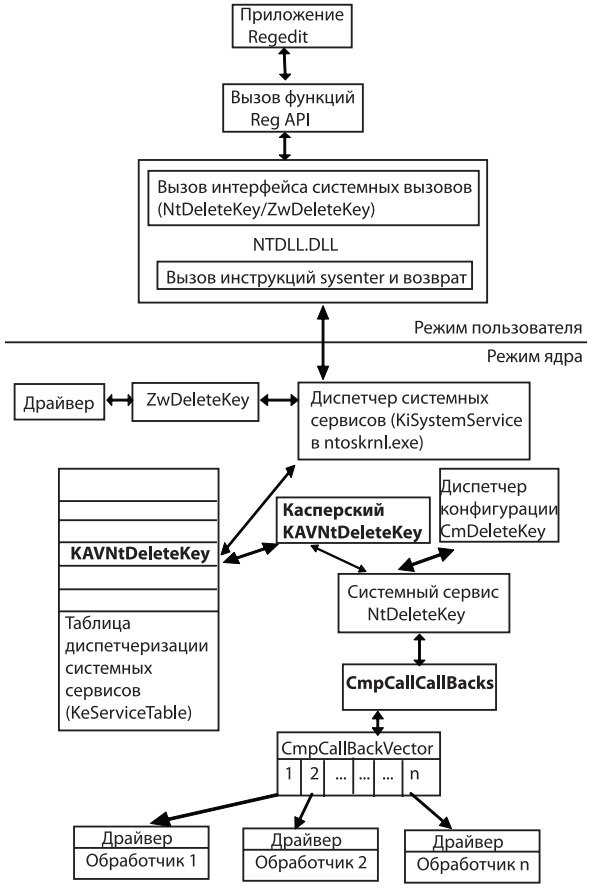


Рис. 5. Обход с помощью механизма обратного вызова реестра, перехваченного антивирусом Касперского системного сервиса удаления ключа

Обход штатного механизма контроля доступа к реестру в ярде ОС Windows XP

Рассмотрим более подробно механизм обратного вызова реестра, архитектура которого была получена в ходе дизассемблирования системных сервисов работы с реестром и диспетчера конфигурации.

Как показано на рис. 4–5, существует некоторый массив CmpCallBackVector, в котором хранятся адреса обработчиков – функции обратного вызова работы с реестром. В переменной CmpCallBackCount хранится количество установленных обработчиков. В ходе выполнения системного сервиса вызывается функция CmpCallCallBacks, которая обращается к этому массиву и поочередно вызывает все установленные обработчики.

Таким образом, контролируя массив обработчиков и их количество, можно контролировать механизм обратного вызова реестра, который, в свою очередь, позволяет осуществлять контроль доступа к реестру на уровне диспетчера конфигурации, компонента исполнительной системы.

Рассмотрим следующую ситуацию: любой модуль уровня ядра имеет доступ во все адресное пространство в ядре. Соответственно, он имеет доступ к массиву, хранящему в себе все адреса обработчиков. Найдя в памяти этот массив (задача, которую можно решить разными способами), любой модуль ядра может обнулить этот массив и переменную, в которой хранится количество установленных в системе обработчиков – функций обратного вызова реестра. Что же тогда произойдет при обращении к реестру? Функция CmpCallBackVector будет вызвана в процессе выполнения системного сервиса, однако, прочитав значение количества установленных обработчиков, которое равно после изменения нулю, просто вернет управление системному сервису. Иначе говоря, не произойдет ничего.

Соответственно, контролируя данный массив и переменную счетчика обработчиков, можно удалять обработчики, установленные средствами обнаружения, и устанавливать средства сокрытия, которые при получении уведомления о доступе к реестру могут осуществлять скрытие в реестре на уровне диспетчера конфигурации.

Заключение

В ходе проведенного исследования были решены поставленные задачи и получены следующие основные результаты:

- описана система доступа к реестру с уровня приложения до уровня ядра;
- описана работа штатного механизма контроля доступа к реестру в ярде ОС Windows XP;
- дано обоснование возможности обхода штатного механизма контроля доступа к реестру в ярде ОС Windows XP.

На основе полученных результатов был сделан следующий вывод: при построении средств контроля доступа к реестру, основанных только на штатном механизме контроля доступа к реестру в ядре ОС Windows XP, контроль доступа гарантировать нельзя, так как в идеологию данного механизма заложена уязвимость, которая связана с невозможностью контроля самого механизма.

Примечания

1 См.: *Холлунг Г., Батлер Дж.* Руткиты: внедрение в ядро Windows. СПб.: Питер, 2007.

2 См.: *Солдатов В.П.* Программирование драйверов Windows. 3-е изд., перераб. и доп. М.: Бином-Пресс, 2006.

3 См.: *Руссинович М., Соломон Д.* Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP и Windows 2000. Мастер-класс. 4-е изд. М.: Русская редакция; СПб.: Питер, 2006.

4 См.: *Орвик П., Смит Г.* Windows driver foundation: Разработка драйверов. М.: Русская редакция; СПб.: БХВ-Петербург, 2008.

5 См.: *Грушо А.А., Шумицкая Е.Л.* Модель невлияния и скрытые каналы // Дискретная математика. 2002. 14:1. С. 11–16.

М.А. Борисов, И.В. Заводцев

ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ОЦЕНКИ УЯЗВИМОСТЕЙ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ

В статье рассмотрено понятие системы оценки уязвимостей и предложен обобщенный подход к оценке выявленных уязвимостей, учитывающий мировой опыт в этом направлении и использующий наработки российских специалистов.

Ключевые слова: автоматизированная система, защищенность информации, система оценки уязвимостей, приоритизация рисков, базовые метрики, контекстные метрики, временные метрики.

Высокий уровень развития средств ведения информационной войны и широкое использование для этого различных видов информационных вторжений определяют потребность в новых подходах, направленных на повышение эффективности систем защиты автоматизированных систем (АС)[1]. В этих условиях важное значение приобретает возможность своевременного выявления и ранжирования IT-персоналом уязвимостей различных программных и аппаратных платформ с целью перераспределения усилий на исправление тех из них, которые представляют наибольшую опасность.

В то же время известные системы оценки уязвимостей (СОУ) наряду с неоспоримыми достоинствами предполагают собственную оценку по разным шкалам[2], что не позволяет свести эти данные воедино для обобщенного анализа. При этом они ориентированы на зарубежное законодательство и не учитывают потребностей российских специалистов (в том числе эксплуатирующих АС в государственных структурах управления).

В связи с этим целесообразным представляется разработка обобщенного подхода к оценке выявляемых уязвимостей, учитывающего, с одной стороны, мировой опыт в этом направлении и предполагающий – с другой, использование наработок российских IT-специалистов.

За основу формируемой СОУ может быть взята открытая схема общей системы оценки уязвимостей (CVSS), которая предоставляет следующие выгоды:

1. Стандартизованная оценка уязвимостей. После нормализации оценок уязвимостей для всех программных и аппаратных платформ возможно использование единой политики управления уязвимостями.

2. Открытость системы, позволяющая каждому пользователю увидеть индивидуальные особенности уязвимости, которые привели к указанной оценке.

3. Приоритизация рисков, что позволяет на основе вычисления контекстной метрики оценивать каждую уязвимость с учетом условий среды конкретной АС, тем самым показывая реальный риск от наличия этой уязвимости в данной организации.

Базовый подход к формированию обобщенной системы оценки уязвимостей предполагает использование трех основных метрик: базовой метрики, временной метрики и контекстной метрики, каждая из которых, в свою очередь, включает соответствующие группы показателей (рис. 1).

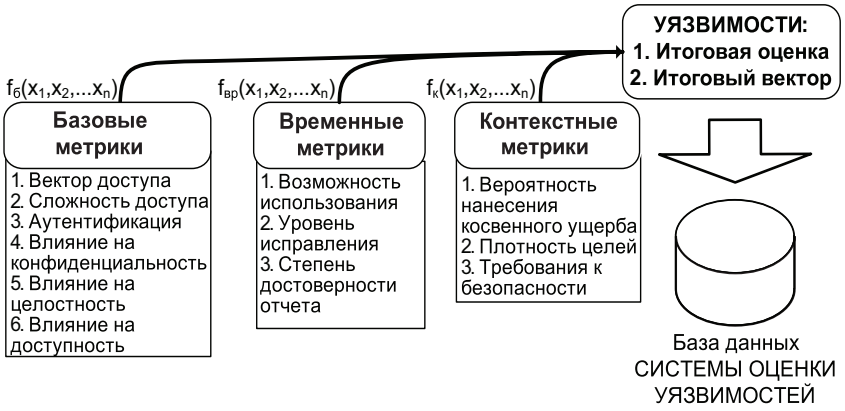


Рис. 1. Структура формирования обобщенной системы оценки уязвимостей

Эти группы метрик описываются следующим образом:

– группа базовых метрик – основные существенные характеристики уязвимости, которые не изменяются со временем и не зависят от среды;
– группа временных метрик – характеристики уязвимости, которые могут измениться со временем, но не зависят от среды;
– группа контекстных метрик – характеристики уязвимости, которые зависят только от среды.

Алгоритм функционирования предлагаемой обобщенной системы оценки уязвимостей можно представить следующей последовательностью действий:

1) составление базовых метрик, позволяющих определить и отобразить основные характеристики выявленной уязвимости;

2) получение подробной информации об уязвимости с учетом среды конкретной АС с использованием временных и контекстных групп метрик, что позволяет обосновать выбор способа минимиза-

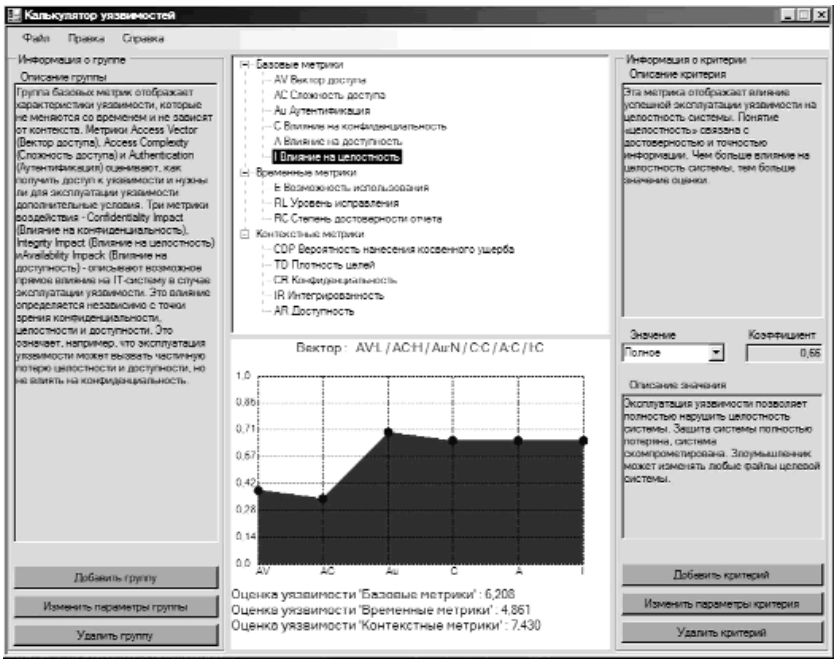


Рис. 2. Экранная форма калькулятора уязвимостей

ции риска от наличия уязвимости. Базовая и временная метрики определяются аналитиками бюллетеней уязвимостей. Контекстная метрика определяется пользователями;

3) после определения базовых метрик на основе базовой формулы вычисляется оценка от 0 до 10 и формируется вектор (рис. 1), олицетворяющий собой открытую архитектуру системы.

Вектор – это текстовая строка, содержащая значения, связанные с каждой метрикой. Каждая метрика в этом векторе представлена сокращенным именем метрики, за которым следует «:» (двоеточие), а затем – сокращенное значение метрики. Вектор содержит последовательность метрик в заранее заданном порядке, при этом символ «/» (слеш) используется для разделения метрик. Если временная или контекстная метрика не используется, то проставляется значение «ND» (не определено). Таким образом, вектор позволяет достаточно точно отобразить методику получения оценки.

В соответствии с этим предложен проект инструментального средства оценки уязвимостей в виде программной реализации алгоритма формирования итоговой оценки и вектора значимости уязвимости (рис. 2).

Примечания

1 См.: Руководящий документ ФСТЭК. Руководящий документ. Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации. М., 1992.

2 См.: Microsoft Corporation. Microsoft Security Response Center Security Bulletin Severity Rating System. November 2002 [cited 16 March 2007] [Электронный ресурс] // Сайт компании Microsoft. [М., 2009]. URL: http://www.microsoft.com/technet/security/bulletin/rating.mspx (дата обращения: 4.11.2009); United States Computer Emergency Readiness Team (US-CERT). US-CERT Vulnerability Note Field De-scriptions. 2006 [cited 16 March 2007] [Электронный ресурс] // Сайт «US-CERT». [М., 2009]. URL: http://www.kb.cert.org/vuls/html/fieldhelp (дата обращения: 4.11.2009); SANS Institute. SANS Critical Vulnerability Analysis Archive. Undated [cited 16 March 2007] [Электронный ресурс] // Сайт SANS Institute. [М., 2009]. URL: http://www.sans.org/newsletters/cva/ (дата обращения: 4.11.2009).

И.В. Шидловский-Москвин

АНАЛИЗ ЗАЩИЩЕННОЙ ИЗОЛИРОВАННОЙ ПРОГРАММНОЙ СРЕДЫ, ОСНОВАННОЙ НА ТЕХНОЛОГИИ ПОЛНОЙ ВИРТУАЛИЗАЦИИ НА ПРИМЕРЕ VMWARE WORKSTATION

Технология виртуализации появилась в 60-х гг. XX в. Однако наибольшее развитие она получила в последние годы, что связано, в первую очередь, с ростом производительной мощности аппаратных средств, простотой и удобством ее использования, а главное – новыми недоступными ранее функциями, такими как одновременная работа нескольких операционных систем (ОС) на одной аппаратной платформе, мигрирование работающих ОС в режиме реального времени между аппаратными платформами и т. д.

Одной из ключевых функций виртуализации в настоящее время является использование ее в качестве механизма безопасности. А именно – создание изолированно программной среды (ИПС), основанной на технологии виртуализации.

Однако многие специалисты переоценивают защищенность самих виртуальных машин (ВМ), а следовательно, построенных на их основе ИПС.

Ключевые слова: виртуализация, типы виртуализации, изолированная программная среда, WinDbg, VMware Workstation, Smartline DeviceLock, SSDT, перехват таблицы диспетчеризации системных сервисов.

В теоретической части данной статьи будет рассмотрена технология виртуализации, типы виртуализации и ИПС, построенные на базе полной виртуализации VMware Workstation, а также теоретическое обоснование возможности проникновения в ИПС. В качестве гостевой и хостовой ОС будет использоваться Microsoft Windows XP. В практической части будут представлено экспериментальное подтверждение описанного в теоретической части предположения. Будет описано два опыта. Первый опыт – изменение адресного пространства процесса гостевой ОС из хостовой на примере процесса

Notepad.exe. Второй опыт – отключение средства контроля доступа пользователей к устройствам (Smartline DeviceLock) в гостевой ОС из хостовой. В заключение будут представлены полученные результаты работы и выводы.

Цели и задачи

Цель данной работы провести анализ защищенности изолированной программной среды, основанной на технологии виртуализации.

Для достижения поставленной цели необходимо решить следующие задачи:

- описать технологию виртуализации;
- сделать обзор известных типов виртуализации;
- описать способ построения ИПС, основанной на технологии виртуализации;
- дать теоретическое обоснование возможности полного доступа в изолированную программную среду, построенную на базе полной виртуализации извне;
- предоставить экспериментальное подтверждение теоретических предположений.

Анализ использованных источников и литературы

При написании работы была использована литература по двум основным направлениям. Первое направление – труды по внутреннему устройству и архитектуре операционных систем. Сюда относятся книга Марка Руссиновича и Дэвида Соломона о внутреннем устройстве Windows, а также книги Эндрю Таненбаума. Второе направление – литература, описывающая технологию виртуализации. В силу доминирования коммерческих разработок в этой области сложно выделить источники, в которых наиболее полно описывается данная технология. Основная информация по данному вопросу была получена с официальных сайтов производителей.

Также следует отдельно выделить статью Ю.К. Сергеева «Использование технологии виртуализации для защиты информации»[1], описывающую идеологию данной работы. В этой статье Сергеев описывает создание механизма безопасности на базе технологии виртуализации и ОС с открытым кодом. Эта работа послужила толчком к написанию данной статьи, которая интересна прежде всего исследованием ОС с закрытым кодом и с использованием коммерческого (т. е. закрытого) продукта виртуализации.

Понятие виртуализации

В широком смысле, понятие *виртуализации* представляет собой сокрытие настоящей реализации какого-либо процесса или объекта от истинного его представления для того, кто им пользуется. Продуктом виртуализации является нечто удобное для использования, на самом деле имеющее более сложную или совсем иную структуру, отличную от той, которая воспринимается при работе с объектом. Иными словами, происходит отделение представления от реализации чего-либо. В компьютерных технологиях под термином «виртуализация» обычно понимается абстракция вычислительных ресурсов и предоставление пользователю системы, которая «инкапсулирует» (скрывает в себе) собственную реализацию. Проще говоря, пользователь работает с удобным для себя представлением объекта, и для него не имеет значения, как объект устроен в действительности.

Виртуализация бывает разная: операционных систем, приложений, систем хранения данных, отдельных аппаратных и программных компонентов вычислительных систем. Пример использования продуктов виртуализации – виртуальная машина Java в браузерах, логические диски в ОС Windows – тоже частный случай виртуализации (ведь на самом деле одно физическое устройство, жесткий диск, представляется пользователю как несколько логических томов).

Но все это было и раньше, почему же в последнее время так много заговорили о виртуализации? А случилось это потому, что за последние несколько лет был совершен большой технологический прорыв в области виртуализации ОС, открывший огромные возможности и перспективы. Под *виртуализацией ОС* понимают процесс создания на физическом компьютере так называемой виртуальной машины (ВМ) (что-то вроде виртуального компьютера), в которой устанавливается своя собственная ОС. Таких ВМ на одной физической платформе может быть несколько, при этом каждая ВМ имеет свои собственные виртуальные аппаратные компоненты: память, процессор, жесткий диск, сетевые адаптеры. Эти ресурсы резервируются ВМ за счет физических ресурсов аппаратного обеспечения компьютера. Такая модель организации вычислительных систем впервые появилась еще в 70-х гг. прошлого века в мэйнфреймах корпорации IBM System 360/370, когда требовалось сохранить предыдущие версии экземпляров ОС. Но лишь в XXI в. эта технология обрела новый смысл на серверных системах и настольных ПК.

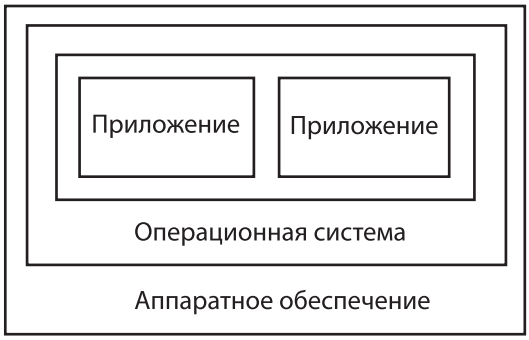


Рис. 1. Классическая архитектура компьютера

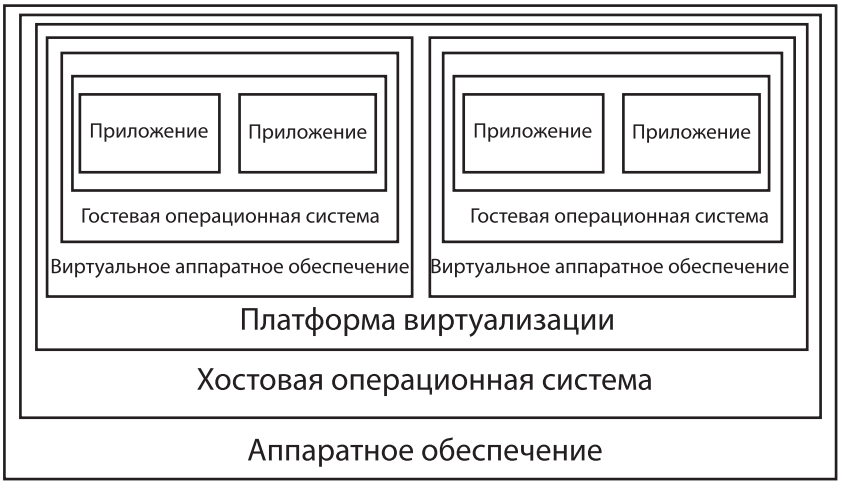


Рис. 2. Один из видов виртуализации ОС

Под виртуализацией ОС будет пониматься следующее: в ОС физического компьютера (хостовой ОС) устанавливается платформа виртуализации (как обычная программа), с помощью которой создаются ВМ. На ВМ устанавливаются различные ОС (гостевые ОС). На рис. 1 и 2 показаны отличия классической архитектуры компьютера от архитектуры, содержащей ВМ.

Теоретическое обоснование опыта № 1

В данной статье будет рассмотрен пример программного продукта на базе технологии *полной виртуализации* VMware Workstation. Гипервизор находится между гостевой ОС и непосредственно оборудованием как слой абстрагирования. Этот слой абстрагирования позволяет любой ОС работать на аппаратных средствах, не имея информации о какой-либо другой гостевой ОС[2].

VMware также виртуализирует в гипервизоре доступные устройства ввода/вывода и соответствующие драйверы для высокоэффективных устройств.

Вся виртуализированная среда сохраняется как файл, и это означает, что система (включая гостевую ОС, гипервизор и конфигурационные файлы ВМ) может быть легко и быстро перенесена на другой сервер для распределения загрузки.

При запуске VMware Workstation 6.0 запускаются несколько процессов:

Имя процесса	Назначение	Владелец
vmnat.exe	NAT	SYSTEM
vmnetdhcp.exe	DHCP	SYSTEM
vmware.exe	Control VMs	user
vmware-authd.exe	Authorization	SYSTEM
vmware-tray.exe	Taskbar	user
vmware-vmx.exe	Image VM	user

Все процессы VMware, за исключением процесса vmware-vmx.exe, являются служебными. Сам же гипервизор и гостевая ОС находятся внутри адресного пространства процесса vmware-vmx.exe.

Рассмотрим схему, представленную на рис. 3, которая описывает принцип построения ИПС на базе технологии виртуализации.

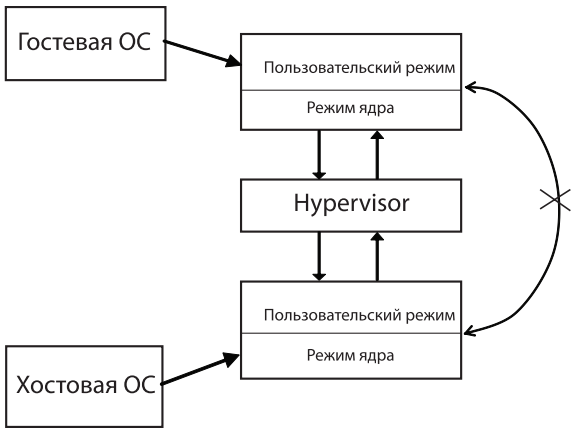


Рис. 3. ИПС на базе виртуализации

В данной схеме предполагается, что весь интерфейс доступа, между гостевой ОС и хостовой ОС, контролируется гипервизором. Гипервизор же в свою очередь эмулирует аппаратный слой для гостевой ОС, что позволяет говорить об одновременной изолированной друг от друга работе двух и более ОС.

Рассмотрим более подробно модель работы процессов ОС. Неважно, о какой ОС (в данном случае о гостевой или хостовой) идет речь. Все процессы в ОС выполняются в пользовательском пространстве. Только служебная информация о процессе – структура процесса, его идентификатор, список открытых дескрипторов и так далее (например, в ОС Windows структура EPROCESS) – находится в режиме ядра, и процесс не имеет к ней непосредственного доступа[3].

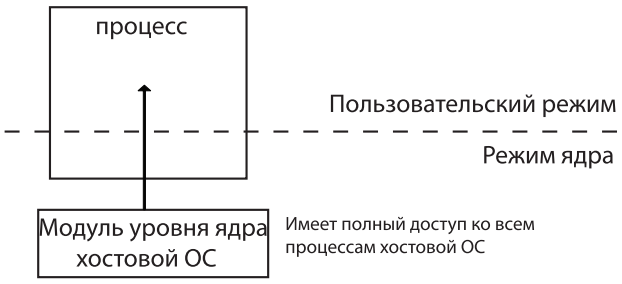


Рис. 4. Адресное пространство процесса в ОС

Однако, как видно из рис. 4, любой модуль уровня ядра имеет полный доступ в адресное пространство процесса. Причем этот доступ не ограничен никакими правами (доступ распространяется на все подсистемы ОС: память, файловую систему, сетевую систему).

Теперь рассмотрим устройство процесса vmware-vmx.exe и его место в хостовой ОС более подробно.

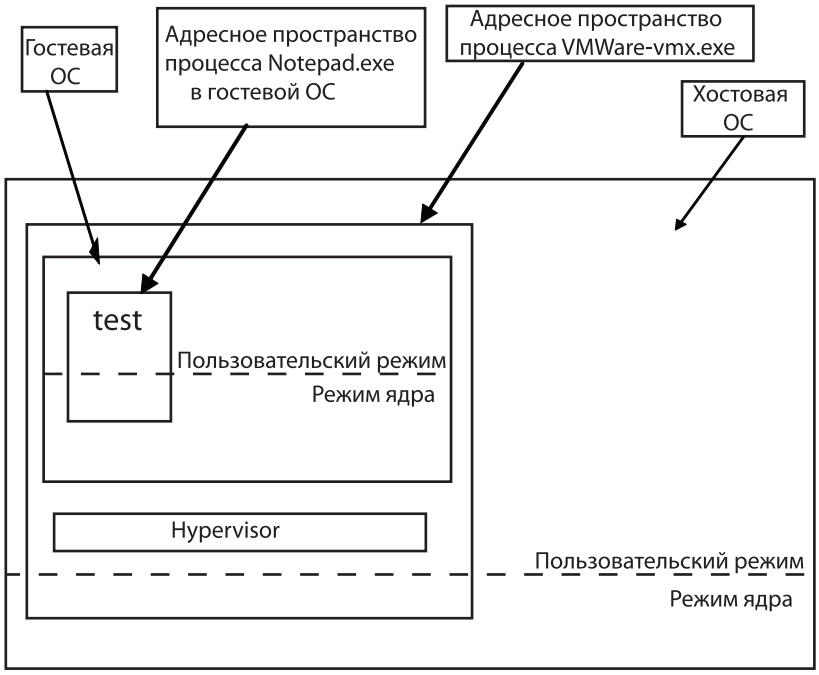


Рис. 5. Адресное пространство процесса vmware-vmx.exe

Как видно из схемы, и гипервизор, и гостевая ОС представляют из себя обычный процесс хостовой системы. Соответственно, как уже говорилось выше, любой модуль уровня ядра хостовой ОС имеет полные права доступа к процессу хостовой ОС, то есть к гипервизору и к образу гостевой ОС, а значит и к самой гостевой ОС.

Рис. 6. Изменение адресного пространства процесса гостевой ОС из хостовой

Рассмотрим данное предположение на примере редактирования текстового файла, т. е. изменения процесса гостевой ОС (notepad.exe) из хостовой ОС с помощью отладчика уровня ядра (WindDbg) – модуля ядра хостовой ОС.

Экспериментальная часть опыта № 1

Суть опыта: попытаться из хостовой ОС в гостевой ОС изменить текстовый документ в режиме реального времени.

Гостевая: С помощью редактора текста Notepad («Блокнот») создаем простой текстовый документ «Текстовый документ.txt» и впечатываем в него слово «hello».

Хостовая: При помощи WinDbg подключаемся к процессу vmware-vmx.exe и осуществляем поиск в виртуальной памяти по шаблону:

0:007> s -u 0x00000000 L?FFFFFFFF "hello"

2bb02340 0068 0065 006c 006c 006f 0440 0438 043c h.e.l.l.o.@.8.<.

Обнаружив адрес искомой строки, производим замену:

0:007> eu 2bb02340 "12"

Гостевая: В блокноте в режиме реального времени строка "hello" изменилась на "12llo".

Вывод: возможно получение полного доступа к пользовательскому процессу в гостевой ОС из хостовой.

Теоретическое обоснование опыта № 2

В опыте № 1 была показана возможность контроля приложения гостевой ОС из хостовой. Теперь продемонстрируем возможность контроля ядра гостевой ОС. Более того, – возможность отключения механизмов безопасности специализированного ПО в области защиты информации на примере средства контроля доступа пользователей к устройствам DeviceLock фирмы Smartline.

Вначале будет описан механизм диспетчеризации, используемый в ОС Windows. Затем будет приведен метод перехвата таблицы системных сервисов SSDT, используемый специализированным ПО по защите информации для обеспечения собственной безопасности. И в конце будет дана экспериментальная часть опыта.

Диспетчеризация системных сервисов

Диспетчеризация системных сервисов начинается с выполнения инструкции *sysenter* (для x86-процессоров Intel Pentium II и выше). Выполнение инструкции приводит к переключению в режим ядра и запуск диспетчера системных сервисов. Номер системного сервиса передается в регистре процессора EAX, а регистр EDX указывает на список аргументов, предоставленных вызвавшим кодом.

Диспетчеризация системных сервисов режима ядра

Как показано на рис. 7, ядро использует номер системного сервиса для поиска информации о нем в *таблице диспетчеризации системных сервисов* (system service dispatch table – SSDT). Каждый элемент этой таблицы содержит указатель на системный сервис[4].

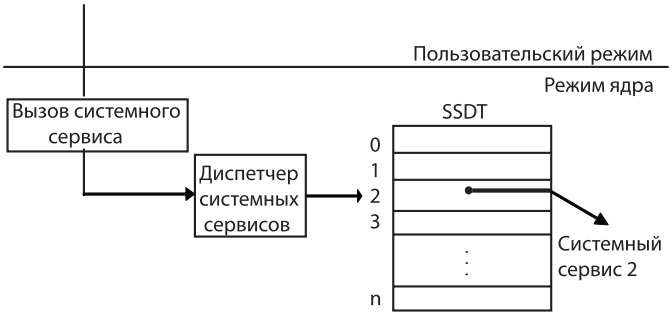


Рис. 7. Вызов системных сервисов

Диспетчер системных сервисов, KiSystemService, копирует аргументы вызвавшего кода из стека потока пользовательского режима в свой стек режима ядра (поэтому вызвавший код не может изменить значения аргументов после того, как они переданы ядру) и выполняет системный сервис.

Инструкции для диспетчеризации сервисов исполнительной системы Windows содержатся в системной библиотеке Ntdll.dll. DLL-модули подсистем окружения вызывают функции из Ntdll.dll для реализации своих документированных функций.

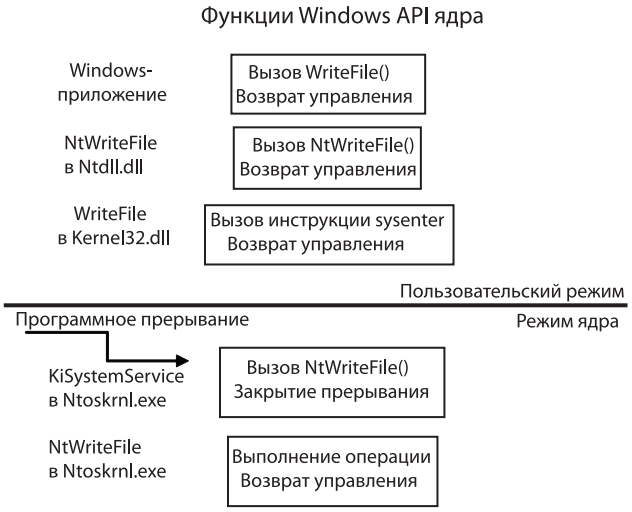


Рис. 8. Диспетчеризация системных сервисов

Как показано на рис. 8, Windows-функция WriteFile в Kernel32.dll вызывает функцию NtWriteFile из Ntdll.dll. Она, в свою очередь, выполняет соответствующую инструкцию (sysenter), вызывающую срабатывание ловушки системного сервиса и передающую номер системного сервиса NtWriteFile. Далее диспетчер системных сервисов (функция KiSystemService в Ntoskrnl.exe) вызывает истинную NtWriteFile для обработки запроса на ввод-вывод.

Перехват таблицы системных сервисов

Классическим методом, используемым антивирусами, специализированными ПО в области защиты информации и средствами контроля доступа, является перехват таблицы системных сервисов. Данный класс программных средств для контроля доступа к реестру, аудиту и контролю процессов системы и обеспечения собственной безопасности перехватывает соответствующие системные сервисы, подменяя их оригинальные адреса адресами своих собственных служебных функций. При обращении диспетчера системных сервисов к соответствующему адресу в таблице SSDT происходит вызов службы специализированного ПО, которая сначала проверяет легитимность вызова, а уже после этого вызывает оригинальный сервис[5].

На рис. 9 представлен перехват таблицы SSDT программным средством контроля доступа DeviceLock.

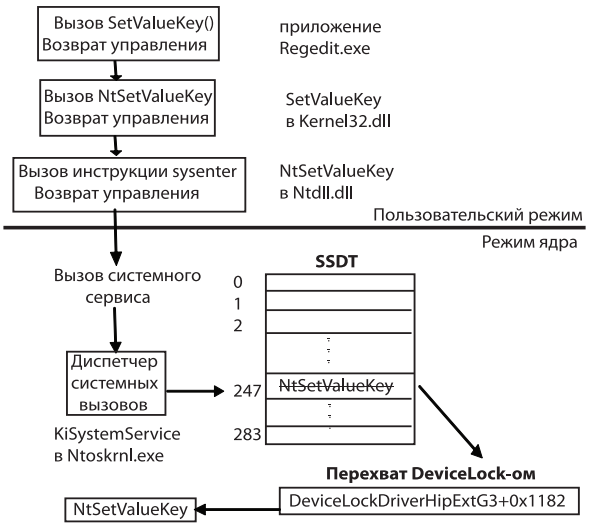


Рис. 9. Перехват DeviceLock-ом системного сервиса NtSetValueKey

Экспериментальная часть опыта № 2
Отключение DeviceLock-а на гостевой ОС

Суть опыта: попытаться из хостовой ОС перехватить системные сервисы таблицы SSDT гостевой ОС и восстановить в ней сервисы, «замененные» DeviceLock-ом на свои. Таким образом, будет осуществлено отключение механизма безопасности (контроль доступа) гостевой ОС из хостовой.

Гостевая:

Устанавливаем DeviceLock. В табл. 1 представлен список перехваченных им системных сервисов, полученный с помощью Rootkit Unhooker.

Таблица 1

Системные сервисы SSDT, перехваченные DeviceLock-ом (Rootkit Unhooker)

Id	Service Name	Hooked	Address (old)	Address
41	NtCreateKey	Yes	~~0x8061A286~~	0xF79F60DE
63	NtDeleteKey	Yes	~~0x8061A716~~	0xF79F615E
65	NtDeleteValueKey	Yes	~~0x8061A8E6~~	0xF79F61C4
108	NtMapViewOfSection	Yes	~~0x805A7480~~	0xF79F62AE
119	NtOpenKey	Yes	~~0x8061B658~~	0xF79F612A
122	NtOpenProcess	Yes	~~0x805C1296~~	0xF79F6278
128	NtOpenThread	Yes	~~0x805C1522~~	0xF79F6242
247	NtSetValueKey	Yes	~~0x8061880C~~	0xF79F6182
257	NtTerminateProcess	Yes	~~0x805C8C2A~~	0xF79F61EE
258	NtTerminateThread	Yes	~~0x805C8E24~~	0xF79F6218

Имя модуля-перехватчика – C:\WINDOWS\system32\drivers**DeviceLockDriverHlpExtG3.SYS**.

Также в таблице представлены исходные адреса оригинальных сервисов.

Гостевая:

Запускаем WinDbg в режиме отладчика ядра (Kernel Debug) на локальной машине (Local).

Отображаем адрес таблицы системных сервисов (SSDT).

```
lkd> dd kiservicetable
80501b8c 80599948 805e6db6 805ea5fc 805e6de8
80501b9c 805ea636 805e6e1e 805ea67a 805ea6be
80501bac 8060bdfe 8060cb50 805e21b4 805e1e0c
80501bbc 805cade6 805cad96 8060c424 805ab5ae
80501bcc 8060ba3c 8059ddbe 805a5a00 805cc8c4
80501bdc 804ff828 8060cb42 8056bcd6 8053500e
80501bec 806050d4 805b1c3a 805eab36 80619e56
80501bfc 805ef028 8059a036 8061a0aa 805998e8
```

Например, 80599948 – адреса 1-го системного сервиса.

В окне Memory выводим на экран содержимое nt!kiservicetable (в удобочитаемом виде Pointer and Symbol) и видим, что это nt!NtAcceptConnectPort

```
80501b8c 80599948 nt!NtAcceptConnectPort
80501b90 805e6db6 nt!NtAccessCheck
80501b94 805ea5fc nt!NtAccessCheckAndAuditAlarm
```

Просмотрев весь список SSDT, выбираем адреса таблицы, указывающие на те системные сервисы, которые подменил DeviceLock (список 1):

```
1)  80501c30 f79f60de DeviceLockDriverHlpExtG3+0x10de
2)  80501c88 f79f615e DeviceLockDriverHlpExtG3+0x115e
3)  80501c90 f79f61c4 DeviceLockDriverHlpExtG3+0x11c4
4)  80501d3c f79f62ae DeviceLockDriverHlpExtG3+0x12ae
5)  80501d68 f79f612a DeviceLockDriverHlpExtG3+0x112a
6)  80501d74 f79f6278 DeviceLockDriverHlpExtG3+0x1278
7)  80501d8c f79f6242 DeviceLockDriverHlpExtG3+0x1242
8)  80501f68 f79f6182 DeviceLockDriverHlpExtG3+0x1182
9)  80501f90 f79f61ee DeviceLockDriverHlpExtG3+0x11ee
10) 80501f94 f79f6218 DeviceLockDriverHlpExtG3+0x1218
```

Список 1. Фрагмент таблицы SSDT после установки DeviceLock-а.

Хостовая:

Структура таблицы диспетчеризации системных сервисов идентична для ОС одной версии. Система на гостевой и хостовой ОС в нашем эксперименте одна и та же – Windows XP. Поэтому, для того чтобы понять, какие именно системные сервисы перехватил DeviceLock, запускаем WinDbg на хостовой системе и получаем их список.

Таблица 2

Название и номера оригинальных сервисов,
перехваченные DeviceLock-ом

№ п/п	№ в SSDT	Название системного сервиса
1	41	NtCreateKey
2	63	NtDeleteKey
3	65	NtDeleteValueKey
4	108	NtMapViewOfSection
5	119	NtOpenKey
6	122	NtOpenProcess
7	128	NtOpenThread
8	247	NtSetValueKey
9	257	NtTerminateProcess
10	258	NtTerminateThread

В адресном пространстве процесса vmware-vmx.exe найдем SSDT-таблицу гостевой системы (т. е в адресном пространстве хостовой ОС найдем адрес SSDT-таблицы гостевой ОС).

Прикрепляемся к процессу vmware-vmx.exe и по шаблону (адрес первого системного сервиса в SSDT-таблице гостевой ОС = = 80599948) ищем саму таблицу диспетчеризации:

```
0:007> s -d 0x00000000 L?FFFFFFFF 80599948
0c981b8c 80599948 805e6db6 805ea5fc 805e6de8 H.Y..m^...^..m^.
```

Таким образом, нашли адрес в SSDT-таблице гостевой ОС в адресном пространстве процесса VMware-vmx.exe (т. е. в адресном пространстве хостовой ОС).

Отобразим адреса, начиная с *0c981b8c*, предположительно – это начало SSDT

```
0:007> dd 0c981b8c
0c981b8c 80599948 805e6db6 805ea5fc 805e6de8
0e2c1b9c 805ea636 805e6e1e 805ea67a 805ea6be
0e2c1bac 8060bdfe 8060cb50 805e21b4 805e1e0c
0e2c1bbc 805cade6 805cad96 8060c424 805ab5ae
0e2c1bcc 8060ba3c 8059ddbe 805a5a00 805cc8c4
0e2c1bdc 804ff828 8060cb42 8056bcd6 8053500e
0e2c1bec 806050d4 805b1c3a 805eab36 80619e56
0e2c1bfc 805ef028 8059a036 8061a0aa 805998e8
```

Гостевая:

```
lkd> dd kiservicetable
80501b8c 80599948 805e6db6 805ea5fc 805e6de8
80501b9c 805ea636 805e6e1e 805ea67a 805ea6be
80501bac 8060bdfe 8060cb50 805e21b4 805e1e0c
80501bbc 805cade6 805cad96 8060c424 805ab5ae
80501bcc 8060ba3c 8059ddbe 805a5a00 805cc8c4
80501bdc 804ff828 8060cb42 8056bcd6 8053500e
80501bec 806050d4 805b1c3a 805eab36 80619e56
80501bfc 805ef028 8059a036 8061a0aa 805998e8
```

Таблица диспетчеризации системных сервисов гостевой ОС найдена.

Теперь восстановим в таблице адреса оригинальных системных сервисов, перехваченных DeviceLock-ом.

Хостовая:

Известно, что DeviceLock перехватывает 247 сервис-таблиц NtSetValueKey (см. табл. 1). Проверим, что лежит по адресу

```
0:007> dd 0c981b8c+(f7*4)
0c981f68 f796e182 80571406 80609092 80522c50
```

Действительно, там лежит адрес функции DeviceLock-а вместо оригинального системного сервиса (см. Список 1, номер 8):

```
8) 80501f68 f79f6182 DeviceLockDriverHlpExtG3+0x1182
```

Гостевая:

Выясним адрес оригинального системного сервиса:

```
lkd> dd nt!ntsetvaluekey
8061880c 58685c6a e8804dfe fff1f6f8 8966f633
```

Хостовая:

Восстановим адрес оригинального системного сервиса гостевой ОС из хостовой:

```
0:007> ed 0c981f68 0x8061880c
```

Гостевая:

Убедимся, что оригинальный системный сервис восстановлен:

```
80501f68 8061880c nt!NtSetValueKey
```

Аналогичным образом восстанавливаем остальные 9 адресов оригинальных сервисов в таблице диспетчеризации гостевой ОС из хостовой.

После восстановления всех перехваченных системных сервисов таблица SSDT выглядит следующим образом.

Таблица 3

Таблица SSDT после восстановления адресов оригинальных сервисов

Id	Service Name	Hooked	Address
41	NtCreateKey	---	0x8061A286
63	NtDeleteKey	---	0x8061A716
65	NtDeleteValueKey	---	0x8061A8E6
108	NtMapViewOfSection	---	0x805A7480
119	NtOpenKey	---	0x8061B658
122	NtOpenProcess	---	0x805C1296
128	NtOpenThread	---	0x805C1522
247	NtSetValueKey	---	0x8061880C
257	NtTerminateProcess	---	0x805C8C2A
258	NtTerminateThread	---	0x805C8E24

Вывод: Возможно получение полного контроля над таблицей системных сервисов SSDT в гостевой ОС из хостовой.

Заключение

В результате проведенного исследования были решены поставленные задачи и получены следующие основные результаты:

- описана технология виртуализации;
- сделан обзор известных типов виртуализации;
- описан способ построения ИПС, основанной на технологии виртуализации;
- дано теоретическое обоснование возможности полного доступа в изолированную программную среду, построенную на базе полной виртуализации извне;
- предоставлено в двух опытах экспериментальное подтверждение теоретических предположений.

На основе полученных результатов были сделаны следующие выводы.

Вывод 1: Возможно получение полного доступа к пользовательскому процессу в гостевой ОС из хостовой.

Вывод 2: Возможно получение полного контроля над таблицей системных сервисов SSDT в гостевой ОС из хостовой.

Примечания

1 *Сергеев Ю.К.* Использование технологий виртуализации для защиты информации // Вестник РГГУ. 2009. № 10. Сер. «Информатика. Защита информации. Математика». С. 98–109.

2 См.: Материалы сайта производителя VMware Workstation [Электронный ресурс] // Сайт VMware. [М., 2009]. URL: http://www.vmware.com/ru/ (дата обращения: 4.11.2009).

3 См.: *Таненбаум Э.* Современные операционные системы. 2-е изд. СПб.: Питер, 2006.

4 См.: *Руссинович М., Соломон Д.* Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP и Windows 2000. Мастер-класс. М.: Русская редакция; СПб.: Питер, 2006.

5 Там же; см. также: *Холлунг Г., Батлер Дж.* Руткиты: внедрение в ядро Windows. СПб.: Питер, 2007.

А.Е. Сатунина, А.С. Сысоев

ПОДХОД К ПРОЕКТИРОВАНИЮ БЕЗОПАСНОСТИ В СЕРВИС-ОРИЕНТИРОВАННЫХ АРХИТЕКТУРАХ

Статья посвящена проблемам обеспечения безопасности функционирования корпоративных информационных систем (КИС) сервис-ориентированной архитектуры (СОА). Рассматриваются подходы к обеспечению безопасности как отдельного вида ИТ-сервиса КИС, обусловленные особенностями информационных систем сервис-ориентированной архитектуры. Перечислены и охарактеризованы процессы обеспечения информационной безопасности, требующие системного анализа и оценок взаимосвязей с другими ИТ-сервисами КИС на всех стадиях жизненного цикла.

Ключевые слова: безопасность корпоративной информационной системы, сервис-ориентированная архитектура (СОА), ИТ-услуги, безопасность в СОА-системах, безопасное управление инфраструктурой, безопасность взаимосвязей между ИТ-сервисами, управление безопасностью.

Одной из актуальных проблем внедрения и функционирования современных корпоративных информационных систем на предприятиях является обеспечение непрерывности бизнеса. Управление непрерывностью бизнеса становится одной из важных задач функционирования корпоративных информационных систем предприятий (КИС)[1]. Современная среда бизнеса – это динамически изменяющаяся среда, поэтому эффективность деятельности предприятий все в большей степени зависит от скорости реакции на изменения требований.

При использовании традиционного подхода к проектированию бизнес-процессов для каждого элемента организационной системы, функционирующего изолированно, строятся модели бизнес-процессов с описанием используемых на каждом шаге ИТ-приложений. При этом каждый бизнес-процесс имеет свои собственные компоненты, реализующие определенные бизнес-действия, которые зачас-

тую совершенно невозможно использовать с небольшими изменениями в других бизнес-процессах и организационных элементах.

В КИС предприятий сервис-ориентированной архитектуры[2] прикладные задачи (функции) предприятия, подлежащие автоматизации, преобразуются в совокупность бизнес-процессов, а бизнес-процессы в свою очередь – в сервисы (ИТ-услуги, которые призвана оказывать создаваемая корпоративная информационная система). Каждый сервис имеет своего заказчика и поставщика и может планироваться и управляться. Выделенные сервисы включают определенные бизнес-процессы, но уже с более гибкой формой поведения, реализации и управления.

Цель сервисного подхода к проектированию бизнес-процессов – создание общей бизнес-логики процессов, что дает возможность повторного использования сервисов, которые могут быть выполнены наиболее соответствующими элементами системы, несмотря ни на какие организационные границы.

Создание гибких бизнес-процессов в КИС предприятий СОА обеспечивается следующими положениями[3]:

- фокус при разработке приложений смещается на выполнение общей бизнес-логики, а не на то, как отдельные компоненты будут взаимосвязаны между собой, т. е. задача заключается в том, чтобы определить, какие сервисы и в какой взаимосвязи наиболее эффективно использовать для реализации бизнес-процессов;
- сервисы могут замещаться эквивалентными сервисами в зависимости от требований;
- сервисы и бизнес-функции могут предоставляться внутренними и внешними поставщиками;
- изменение формы соединения сервисов: переход от прямого соединения между сервисами к соединению через промежуточный слой – сервисную инфраструктуру, что повышает гибкость соединений между сервисами.

Для ИТ-приложений, построенных согласно СОА принципам, безопасность функционирования корпоративной информационной системы становится еще более критичной. Реализация каждого ИТ-приложения в рамках всей системы предусматривает выполнение соединений часто между несвязанными сервисами и ИТ-приложениями, относящимися к различным организационным структурам. Подобная среда нередко является достаточно незащищенной, уязвимой и ограниченной с точки зрения безопасности. В этом случае система безопасности в КИС-предприятиях может быть представлена как многоуровневая система: безопасность на уровне отдельного сервиса, на уровне сервисной инфраструктуры, на уровне взаимодействия сервисов.

Рис. 1. Обобщенная схема уровней безопасности сервисов в ИС сервис-ориентированной архитектуры

Защита информации – вопрос комплексный, предусматривающий решение следующих задач[4]:

– идентификация клиента (пользователя либо агента) в системе;
– проверка прав доступа к защищаемым данным в рамках общезначимого (системного) контекста безопасности;
– шифрование трафика между клиентами и сервисами информационной системы;
– обеспечение целостности данных.

Безопасность в СОА достаточно хорошо структурирована на уровне описания сервисов. На уровне реализации она реализуется тем же образом, что и любая безопасность в любой другой архитектуре, поэтому «опасность» серверов может быть равна «опасностям» способов реализации[5]. То есть «опасности», например цифровой подписи, зависят от алгоритма подписи, а не от архитектуры самой СОА.

При разработке безопасности сервисов необходимо учитывать следующие аспекты:

– безопасность в СОА-системах должна быть основана не только на соблюдении моделей, установленных в сценариях и функциональных требованиях, но и должна динамически перестраиваться в соответствии с изменяемыми требованиями, то есть модели безопасности должны быть взаимосвязаны с требованиями;
– необходимо иметь динамические доверенности (обязательства) для взаимосвязей с партнерами, заказчиками, клиентами, служащими, основанные на опыте взаимодействия с ними в длительный период времени;

– сфера ответственности не должна жестко регламентировать правила поведения внутри сервисов, но должна быть способна обеспечивать жесткие требования безопасности через политику управления инфраструктурой и взаимосвязями между сервисами.

Для создания безопасного управления инфраструктурой и взаимосвязями между сервисами необходимо описать и проанализировать методы и технологии, связанные с реализацией следующих процессов:

– идентификация пользователей и сервисов;
– подключение к ИТ-приложениям других организаций на основе транзакций в режиме реального времени;
– функционирование композитных приложений (контроль безопасности должен выполняться как на уровне отдельного сервиса, так и на уровне комбинаций сервисов);
– управление безопасностью через цепочки различных ИТ-приложений;
– защита данных при передаче и хранении;
– соответствие изменяющихся требований корпоративным, отраслевым и государственным стандартам и регламентам;
– управление безопасностью на всех этапах жизненного цикла создания сервисов.

Идентификация пользователей и сервисов играет ключевую роль при переходе к сервисным архитектурам.

Например, пользователь, работая в корпоративной ИТ-среде, использует различные ID: Desktop ID, Notes ID, SAP ID, VPN ID, Corporate Travel ID, Online Bank Account, Financial Services Account, Provider ID и др.

В этом случае возникает ряд проблем:

– каждое приложение привносит свой собственный ID;
– каждый ID не работает с другим ID;
– каждый ID повышает сложность идентификации;
– каждый ID добавляет соответствующие бизнес-риски.

В СОА идентификация сервисов выполняется в инфраструктуре, которая реализует посреднические функции, чтобы сервисы могли легко соединяться друг с другом без опасений о том, как представить и передать пользовательские идентификационные данные от одного сервиса к другому.

Очень маловероятно, что идентификация пользователей будет выполняться одинаково во всех сервисных компонентах в течение бизнес-процесса, который охватывает различные организации. Поэтому необходимо использование идентификационных сервисов, которые смогут вычислить идентификаторы обращающегося поль-

зователя (сервиса), подтвердить, что он авторизован выполнять запрашиваемые операции, и передать эти данные целевому сервису. Целевой сервис может понять и использовать идентификационные данные пользователя или сервиса, выполняющего запрос.

Управление безопасностью пронизывает все аспекты жизненного цикла сервис-ориентированного приложения.

Фаза «Моделирование» – определение корпоративной политики безопасности, построение модели требований безопасности и модели безопасности приложений.

Фаза «Проектирование/разработка» – определение политик безопасности приложений, реализация и тестирование безопасности приложений.

Фаза «Развертывание» – конфигурирование инфраструктуры для обеспечения безопасности приложений; безопасная интеграция людей, процессов и информации.

Фаза «Управление» – управление безопасностью приложений, управление идентификацией, мониторинг за реализацией требований безопасности в ходе выполнения процессов.

В информационных системах, построенных на принципах СОА, присутствуют проблемы обеспечения безопасности сервисов, к которым применяются практически все ее составляющие: идентификация, аутентификация, авторизация, конфиденциальность, целостность, аудит и согласование, политики управления и доступности.

Примечания

1 *Задорожный В.* Управление непрерывностью бизнеса – это не процесс, это культура // Intelligent Enterprise / Russian Edition. Корпоративные системы. 2008. № 16 (191). С. 30.

2 См.: Understanding SOA Security // IBM International Technical Support Organization. 2007.

3 *Шепелявый Д.* Information Security // Информационная безопасность. 2008. № 1. С. 28–29.

4 Там же.

5 Там же.

М.А. Борисов

К ВОПРОСУ О МОДЕЛИРОВАНИИ СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ В УСЛОВИЯХ ИНФОРМАЦИОННОГО ПРОТИВОБОРСТВА

В статье рассмотрено понятие защищенности информации и предложен вариант базовой модели по оценке системы защиты информации в условиях информационного противоборства, действующей на основе анализа информационных рисков и расчете финансово-экономических показателей.

Ключевые слова: защищенность информации, информационное противоборство, система защиты информации, ценность информации.

Применительно к защите информации и – как результату защиты – защищенности информации системный подход предполагает исследование конфликта целей сторон информационного противоборства, факторов, влияющих на выигрыш той или иной стороны, и формализацию данного влияния.

В условиях информационного противоборства, с одной стороны, на информацию посредством нападения воздействует противник с целью ее разрушения (модификации, уничтожения), а с другой стороны – противодействует система защиты.

Защищенность информации также может изменяться с течением времени и является свойством системы защиты информации (СЗИ) достигать целевого эффекта при взаимодействии с системой информационного нападения (СИН). При этом целевым эффектом СЗИ является та или иная степень защищенности информации, измеряемая соответствующей математической мерой. Интегральный количественный показатель защищенности Z должен также носить вероятностный (субъективный или объективный) характер и показывать выигрыш системы защиты на заданном временном интервале $[0, T]$.

Выигрыш СЗИ зависит от двух компонент: П – показателя полноты учета возможных стратегий нападения, на противостояние которым нацелена СЗИ при ее разработке, и R – показателя эффективности применения конструктивно заложенных в СЗИ стратегий защиты на интервале $[0, T]$. Тогда показатель защищенности определяется как: $Z(T) = \Phi[\Pi, R(T)]$.

Обобщенная модель информационного противоборства включает в себя модели: СИН, СЗИ, разведки, конкретного способа информационного нападения, конкретного способа информационного противоборства, систем управления как нападением, так и защитой, а также средства защиты и информационного нападения.

В ходе информационного противоборства выбирается стратегия нападения и защиты, она может быть чистая или смешанная. При чистой стратегии одна сторона знает конкретный вариант защиты (нападения), в то время как другая сторона осуществляет выбор защиты (нападения) из определенного множества. При смешанной стратегии альтернативы выбора являются случайными.

Задача системы управления в ходе информационного противоборства заключается в оценке защищенности информационно-вычислительной системы (ИВС) и принятии решения по повышению ее защищенности. Таким образом, видна необходимость создания такой гибкой модели СЗИ, действующей на основе анализа информационных рисков и расчете финансово-экономических показателей, которая удовлетворяла бы следующим требованиям:

– учет и классификация всех возможных угроз;
– оценка рисков нанесения возможного ущерба;
– определение оптимального набора элементов защиты.

В качестве оптимального понимается такой набор элементов защиты, при котором система находилась бы в точке равновесия, в предложенной модели точка равновесия – это минимально возможный риск при минимально возможных затратах на СЗИ.

В качестве примера рассмотрим следующий фрагмент ИВС (рис. 1) с условиями:

– воздействие на ИВС (угроза) является внешней, то есть воздействие осуществляется только через шлюз;

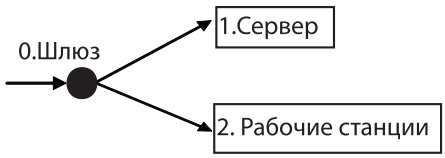


Рис. 1. Фрагмент информационно-вычислительной системы

– данная ИВС может быть усложнена различными элементами, присоединяемыми к шлюзу;
– любой присоединяемый элемент должен являться менее значимым, чем сервер.

В данном фрагменте ИВС каждый элемент имеет две характеристики – вероятностную и ценностную.

Ценностная характеристика заключается в том, что у каждого элемента есть четыре состояния:

0 – базовое (чистая ценность),

1 – с одним улучшением (чистая ценность + ценность одного улучшения),

2 – с двумя улучшениями (чистая ценность + ценность двух улучшений),

3 – с тремя улучшениями (чистая ценность + ценность трех улучшений).

Вероятностная характеристика заключается в том, что методом экспертной оценки вводится вероятность успешной атаки на элемент и соответствующие вероятности.

После улучшения ИВС предполагается, что вероятность успешной атаки должна падать, в то же время предложенная модель также устойчива и к таким улучшениям, которые приводят к увеличению вероятности успешной атаки.

В предлагаемой модели набор и характеристика угроз определяются пользователем, то есть эффективность атаки определяется экспертным путем указания определенного значения или диапазона значений (табл. 1), при этом список угроз и их эффективность являются редактируемыми.

Таблица 1

Вариант диапазона значений эффективности атаки

Угрозы ИБ	Шлюз	Сервер	Рабочие станции
Хищение	50–100%	80–100%	10–30%
Утрата	60–90%	70–100%	10–20%
Блокирование	40–70%	80–100%	10–20%
Уничтожение	80–100%	100%	20–40%
Модификация	40–70%	80–100%	10–20%
Отрицание подлинности	30–50%	50–90%	10–20%
Навязывание ложной информации	40–70%	80–100%	10–30%

В предлагаемой модели для каждой угрозы составляется функция потерь, это некое подобие математического ожидания потерь:

$$X_N = P_{0n} \cdot C_{0n} \cdot F_0 + P_{0n} \cdot P_{1k} \cdot C_{1k} \cdot F_1 + P_{0n} \cdot P_{2t} \cdot C_{2t} \cdot F_2 ,$$

где X – функция потерь; C_{0n} – ценность 0-элемента в n-состоянии; F – эффективность атаки; P_{1k} – вероятность успешной атаки на 1-элемент в k-состоянии $n, k, t = (0, 1, 2, 3)$; N – количество всевозможных комбинаций элементов системы, в данном случае $4^3 = 64$.

Представленная функция имеет ряд ограничений:

– функция не учитывает ограниченность в финансовых ресурсах;
– min(X) можно получить только со всеми улучшениями;
– в подобных моделях оптимальное решение, как правило, лежит где-то посередине (рис. 2), т. е. оптимальное решение – это точка равновесия, или точка глобального минимума;
– модель носит дискретный характер, и переход к непрерывной модели на данном этапе не представляется возможным.

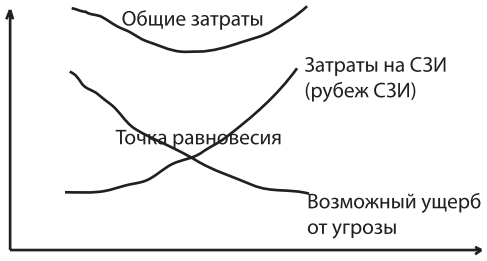


Рис. 2. График нахождения оптимального решения

Нахождение оптимального значения функции потерь для каждой угрозы можно решить тремя методами.

Первый метод:

1. Найдем все возможные значения X.

2. Проверим min(X) на оптимальность, учитывая ограничения на финансы.

3. Если оптимальное решение не найдено, то найдем математическое ожидание X, в данном случае оно совпадает со средним арифметическим, то есть $\underline{X} = (\sum X_R)/R$, по всем R = 1...N.

4. Выберем все X в некотором диапазоне от $\underline{X}$.

5. При использовании данного метода осуществляется уход от рисковых ситуаций. Для устранения этой проблемы необходимо использовать функцию $\mathrm{mod}X_N$.

Второй метод:

1. Найдем все возможные значения X.

2. Проверим $\min(X)$ на оптимальность, учитывая ограничения на финансы.

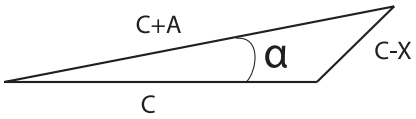
3. Если оптимальное решение не найдено, то вычислим $\underline{X} = \mathrm{mod}X_S$, $S = (1...N)$.

4. Выберем все X в некотором диапазоне от $\underline{X}$.

5. Если решений окажется несколько, необходимо выбрать то, при котором достигается минимальная ценность СЗИ.

Третий метод – метод визуального сравнения.

Порой даже после того как модель выдает оптимальное решение, могут найтись такие параметры системы, при которых она будет более оптимальной. Поэтому в качестве сравнения предлагается метод треугольника, который, конечно, можно автоматизировать и применять как отдельный метод для нахождения оптимального X, но и он не лишен своих недостатков.



C – базовая ценность системы;

A – ценность улучшения;

X – возможные потери.

Далее, используя теорему косинусов, ищем угол α,

$$(C + X)^2 = C^2 + (C + A)^2 - 2\cdot(C + A)\cdot A\cdot\cos(\alpha).$$

Далее задача оптимизации сводится к нахождению $\max(\alpha)$.

Безусловно, представленная модель не охватывает всех условий, необходимых для создания совершенной СЗИ, но по крайней мере она может использоваться как некий базовый элемент более общей модели или послужить плацдармом для создания более общей модели.

А.С. Комаров

ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ В ПОЧЕРКОВЕДЕНИИ: ПРОГРАММНАЯ РЕАЛИЗАЦИЯ*

В статье описывается программный продукт для реализации интеллектуального анализа почерковедческих данных, предназначенный для почерковедческих исследований и поддержки почерковедческой экспертизы. Интеллектуальный анализ данных опирается на ДСМ-метод автоматического порождения гипотез. Описываемый программный продукт позволяет группе исследователей совместно работать с системой. Указаны задачи, решаемые в настоящий момент и планируемые для решения в дальнейшем.

Ключевые слова: интеллектуальный анализ данных, ДСМ-метод, почерковедческая экспертиза, почерк, веб-приложение, трехуровневая клиент-серверная архитектура.

В настоящее время почерковедческая экспертиза, призванная помочь при расследовании преступлений, находится в кризисе. Это связано в первую очередь с тем, что эксперт-почерковед при решении идентификационной задачи почерковедческой экспертизы или установлении каких-либо атрибутов исполнителя рукописи (пол, возраст и т. п.), а также при решении атрибутивно-диагностических задач традиционно опирался на соответствующие методики. Создание таких методик – длительный и трудоемкий процесс. Основные методики были созданы во второй половине прошлого века. В их создании принимали участие несколько научно-исследовательских институтов. Методики опирались на прописи, использовавшиеся для обучения детей письму в школе.

* Работа выполнена при поддержке РФФИ, проект № 09-07-0087-а «Интеллектуальная система анализа криминалистических данных».

В настоящее время прописи того образца устарели, а создание новых методик на основе современных стандартов прописей не финансируется. Поэтому возникает потребность в поиске новых подходов к созданию инструментов для почерковедческой экспертизы.

Методы, применяющиеся для решения задач почерковедческой экспертизы, в основном носят вероятностно-статистический характер (графологические методы не рассматриваются, так как они не имеют строгого научного обоснования). Исключение составляет метод фазового анализа[1], а также количественные методы криминалистического исследования кратких записей, выполненных намеренно измененным почерком скорописным способом, основанные на теории кубического сплайна, разработанные Е.В. Яковлевой[2].

В работе В.Ф. Орловой[3] утверждается, что зависимости между исполнителем рукописи и его почерком носят принципиально вероятностно-статистический характер. Однако результаты исследования связей между факторами, влияющими на почерк, и признаками почерка[4] позволяют предположить, что детерминистская составляющая в этих зависимостях присутствует.

Такое расхождение в суждениях объясняется большой сложностью задачи, вызванной многокомпонентностью в описании исполнителя рукописи и множественностью факторов, влияющих на формирование почерка. Подобная многокомпонентность и многофакторность проявления особенностей если и не исключает полностью возможности существования детерминированных связей, то сильно затрудняет, а иногда и делает практически неосуществимым их выявление. Однако из этого еще не следует, что связи носят статистический характер и могут быть обнаружены только вероятностно-статистическими методами.

Очевидно, что поиск детерминированных связей является крайне трудоемким процессом, который не под силу человеку без применения компьютерных методов.

Существующие компьютерные программы, которые используются в этой области в настоящее время, можно разделить на две категории по своей функциональности:

– автоматизирующие процесс вычисления;

– прогнозирующие ответ задачи.

Программы первой категории носят исключительно прикладной характер. Они представляют собой некий программный интерфейс, в который заложены определенная методика и алгоритм вычисления, и предназначены исключительно для автоматизации вычислений, связанных с этой методикой. Без заранее разработанных методик существование таких программ невозможно.

К программам второй категории относится система, разрабатываемая в настоящий момент в МИФИ[5]. В основе этой системы лежат нейронные сети, что позволяет ей не только предсказывать правильный ответ задачи, но и обучаться на примерах. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными и выходными данными, а также выполнять обобщение. В системах такого рода методики могут использоваться при проектировании нейронной сети, но не являются необходимым элементом при создании системы, в отличие от систем первой категории.

Все эти программы объединяет одно: прогнозируя или вычисляя результат задачи, они не вскрывают зависимостей в системе «человек–рукопись», а для дальнейших исследований в области почерковедения требуется именно это. Экспертам-почерковедам необходимы новые подходы, применяя которые можно было бы создавать новые или модернизировать старые методики, анализируя выявленные зависимости. В данной статье рассматривается программная реализация одного из таких подходов – интеллектуального анализа почерковедческих данных.

Интеллектуальный анализ данных позволяет выявить зависимости в системе «человек–почерк», уточнить модель предметной области и подсказать подходы к решению некоторых конкретных задач. Этот анализ осуществляется с помощью ДСМ-метода автоматического порождения гипотез[6]. Дадим краткое описание этого метода[7].

ДСМ-метод автоматического порождения гипотез для баз данных (фактов) с неполной информацией является логико-комбинаторным средством анализа структурированных данных и порождения гипотез о причинно-следственных зависимостях, неявно содержащихся в базах фактов (БФ). Гипотезы о возможных причинах изучаемых эффектов являются средствами для прогнозирования этих эффектов.

ДСМ-метод образован четырьмя компонентами:

1) автоматизированные правдоподобные рассуждения;

2) аксиоматизированные условия их применимости к БФ определенного типа;

3) представление знаний в виде квазиаксиоматических (открытых) теорий, лишь частично характеризующих предметную область. Квазиаксиоматические теории состоят из аксиом предметной области, аксиом структуры данных, аксиом, представляющих процедуры и правила правдоподобного вывода (индукции, аналогии, абдукции);

4) интеллектуальные системы типа ДСМ, состоящие из решателя задач, базы фактов и базы знаний и специального интерфейса.

Решатель задач типа ДСМ содержит *рассуждатель*, реализующий формализованные эвристики анализа данных в БФ. Эти эвристики являются правдоподобными рассуждениями, которые осуществляют синтез познавательных процедур:

- эмпирической индукции – порождения из БФ, содержащей позитивные (+)- и негативные (–)- примеры, гипотез о (+)- и (–)- причинах изучаемого эффекта;
- аналогии – предсказания наличия (отсутствия) этого эффекта в случаях неопределенности посредством гипотез о (+)- и (–)- причинах;
- абдукции – объяснения начального состояния БФ посредством (+)- и (–)- гипотез либо для принятия гипотез на достаточном основании, либо расширения БФ, либо для констатации «практической расходимости» процесса предсказания.

ДСМ-метод реализует важный принцип: «сходство объектов определяет сходство эффектов и их повторяемость». Это отличает его от статистических методов и делает инструментом формализованного качественного анализа данных.

ДСМ-метод решает определенный класс задач, называемых ДСМ-задачами, и дает положительный результат только при условии хорошо разработанной предметной области.

Предварительный анализ задач почерковедческой экспертизы показал их принадлежность к классу ДСМ-задач. Но данная предметная область крайне сложна, и построение удачной модели с первого раза не представляется возможным. По этой причине проблема адаптации ДСМ-метода для решения задач почерковедческой экспертизы носит итерационный характер, где на каждом этапе решается вопрос необходимости изменения объектной модели и пополнения базы данных.

В итоге перед разработчиками системы интеллектуального анализа почерковедческих данных встала задача создания программы, позволяющей хранить данные, работать с ними и проводить их интеллектуальный анализ с применением ДСМ-метода. Кроме того, подразумевается, что с одним массивом данных должны иметь возможность работать как разработчики системы, так и эксперты-почерковеды, то есть группа, члены которой могут выполнять разные операции в зависимости от своих прав доступа. Таким образом, система должна поддерживать авторизацию пользователей и обеспечивать одновременный доступ с разных компьютеров в независимости от их местонахождения и установленного программного обеспечения. Важным фактором также является нормальное функционирование системы на слабых компьютерах при условии трудоемких вычислений.

В ходе анализа поставленной задачи была выбрана реализация данной системы в виде веб-приложения с трехуровневой архитектурой. Подробнее трехуровневая архитектура и ее реализация в данной системе будет описана далее, сейчас рассмотрим принцип работы веб-приложения.

Под веб-приложением понимается клиент-серверное приложение, в котором в роли клиента выступает браузер, а в роли сервера приложений – веб-сервер. Веб-приложение получает запрос от пользователя, выполняет вычисления, после этого формирует веб-страницу и отправляет ее пользователю по сети с использованием протокола HTTP. Логика приложения сосредоточивается на сервере, а функция браузера заключается в основном в отображении информации, загруженной по сети с сервера, и передаче обратно данных пользователя. При этом вся основная вычислительная нагрузка ложится на сервер, предъявляя высокие требования к компьютеру, на котором он находится. Единственным требованием к компьютеру пользователя остается наличие на нем установленного браузера и подключения к одной с сервером сети. Браузер, как правило, входит в состав каждой операционной системы. Функции его обновления и сопровождения лежат на поставщике операционной системы и никак не связаны с разработчиком самого веб-приложения. Все браузеры способны отображать веб-страницы и обладают схожим функционалом, что обеспечивает независимость веб-приложения от конкретной операционной системы конечного пользователя и делает его межплатформенным сервисом. Кроме того, современные технологии позволяют создавать веб-приложения, способные взаимодействовать с пользователем в интерактивном режиме, практически стирая грань между веб-приложениями и локальными приложениями.

В отличие от классического устройства клиент-серверного приложения в данной системе хранение данных вынесено на отдельный уровень, что соответствует принципам трехуровневой архитектуры, которая предполагает наличие следующих компонентов приложения: клиентское приложение, подключенное к серверу приложений, который в свою очередь подключен к серверу базы данных. Рассмотрим структуру системы подробнее.

Под клиентским приложением (или терминалом) понимается интерфейсный компонент, который представляет первый уровень – собственно приложение для конечного пользователя. Первый уровень не должен иметь прямых связей с базой данных и не должен быть нагружен бизнес-логикой. (Здесь и далее под бизнес-логикой понимается реализация правил и ограничений автоматизируемых операций. Синонимом является термин «логика предмет-

ной области». При разработке системы бизнес-логика реализуется в классах и методах классов в случае использования объектно-ориентированных языков программирования или процедурах и функциях в случае применения процедурных языков.) На первый уровень могут быть вынесены и обычно выносятся простейшие фрагменты бизнес-логики: интерфейс авторизации, алгоритмы шифрования, проверка вводимых значений на допустимость и соответствие формату, несложные операции (сортировка, группировка, подсчет значений) с данными, уже загруженными на терминал. Клиентское приложение со столь ограниченным функционалом еще принято называть «тонким клиентом». В нашем случае роль тонкого клиента, как уже было сказано, может выполнить любой имеющийся на компьютере конечного пользователя браузер. Бизнес-логика на этом уровне в рассматриваемой системе включает в себя проверку вводимых пользователем значений и операции с полученными с сервера данными. Она реализована на языке JavaScript – скриптовом языке, который используется при создании сценариев поведения браузера, встраиваемых в веб-страницы, – при использовании фреймворка jQuery.

Сервер приложений располагается на втором уровне. На этом уровне сосредоточена бóльшая часть бизнес-логики, отвечающая за реализацию правил взаимодействия объектов предметной области. Вне его остаются фрагменты, экспортируемые на терминалы, а также погруженные в третий уровень хранимые процедуры и триггеры. В описываемой системе на этом уровне расположена часть бизнес-логики, связанная с обработкой и анализом данных, авторизацией пользователей и разграничением их прав. Она реализована на языке PHP версии 5. PHP – скриптовый язык, предназначенный для генерации HTML-страниц на веб-сервере и работы с базами данных. В качестве сервера приложений в данной системе выступает HTTP-сервер Apache версии 2.2, поддерживающий исполнение скриптов, написанных на языке PHP.

Сервер базы данных обеспечивает хранение данных и выносится на третий уровень. Обычно это стандартная реляционная или объектно-ориентированная система управления базами данных (СУБД). Если третий уровень представляет собой базу данных вместе с хранимыми процедурами, триггерами и схемой, описывающей приложение в терминах реляционной модели, то второй уровень строится как программный интерфейс, связывающий клиентские компоненты с прикладной логикой базы данных. В данной системе в качестве СУБД используется MySQL Community Server версии 5.1. Объектная модель разрабатываемой системы реализована в рамках схемы базы данных.

В сравнении с приложениями, обладающими двухуровневой клиент-серверной или файл-серверной архитектурой (приложения с файл-серверной архитектурой схожи по своей структуре с локальными приложениями и используют сетевой ресурс только для хранения программы и данных; функция сервера – хранение данных и кода приложения, вся обработка данных происходит исключительно на стороне клиента), можно выделить следующие достоинства приложений с трехуровневой архитектурой:

- масштабируемость – способность системы увеличивать свою производительность при добавлении ресурсов (обычно аппаратных);
- конфигурируемость – изолированность уровней друг от друга позволяет (при правильном развертывании архитектуры) быстро и простыми средствами переконфигурировать систему при возникновении сбоев или при плановом обслуживании на одном из уровней;
- высокая безопасность;
- высокая надежность;
- низкие требования к скорости канала (сети) между терминалами и сервером приложений;
- низкие требования к производительности и техническим характеристикам терминалов.

Недостатки таких приложений вытекают из их же достоинств:

- более высокая сложность создания приложений;
- сложнее в разворачивании и администрировании;
- высокие требования к производительности сервера приложений и сервера базы данных, а значит, и высокая стоимость серверного оборудования;
- высокие требования к скорости канала (сети) между сервером базы данных и сервером приложений.

Таким образом, выбранная реализация полностью удовлетворяет предъявленным к системе требованиям. Единственным недостатком при эксплуатации данной системы остаются высокие требования к серверной части: необходимость как минимум одного мощного компьютера и сложность в разворачивании и администрировании на нем системы.

В настоящий момент с системой работает небольшая группа людей, что позволяет использовать конфигурацию системы, когда физически сервер приложений совмещен с сервером базы данных на одном компьютере, к которому по сети могут подключаться один или несколько терминалов. Это частично сглаживает недостатки архитектуры, снимая требование к скорости канала между серве-

ром базы данных и сервером приложений и понижая требования к производительности серверов. Подобная конфигурация применима и в случае развертывания системы в компьютерном классе. В то же время при увеличении нагрузки на систему всегда можно разнести сервер приложений и сервер баз данных на разные компьютеры, не теряя работоспособности системы и не изменяя программного кода.

Как говорилось выше, решение задачи интеллектуального анализа криминалистических данных с применением ДСМ-метода носит итерационный характер. При этом идет постоянный процесс модернизации существующей системы.

В первую очередь в системе реализован функционал, связанный с доступом к данным и их хранением. Пользователи могут вводить, просматривать и редактировать имеющуюся в базе информацию.

В основу схемы базы данных системы заложена объектная модель предметной области. К основным объектам модели относятся люди (исполнители рукописей) и документы (рукописи). Для каждого человека содержится информация о его возрасте и поле. Каждый документ представляет собой описание рукописного документа в терминах языка описания общих признаков почерка. Для каждого человека хранится связь с документом, исполнителем которого он является. На данный момент в единой базе хранятся подготовленные экспертами-почерковедами сведения о более чем 250 образцах почерка.

Кроме того, в рамках системы реализован алгоритм интеллектуального анализа данных на основе модифицированного ДСМ-метода. Применение данного алгоритма к имеющимся данным позволяет проводить эксперименты по выявлению гипотез о зависимостях между полом исполнителя и признаками его почерка и между разными группами признаков почерка. Совместный с экспертами-почерковедами анализ полученных гипотез показал недостаточность описания почерка только общими признаками для решения задачи определения и, как следствие, необходимость расширения объектной модели за счет введения новых параметров описания как самих документов, так и их исполнителей[8].

В настоящий момент ведутся исследования возможности расширения объектной модели за счет добавления описания психологических свойств человека и частных признаков почерка и адаптации алгоритмов для их анализа. Также рассматривается вопрос о целесообразности включения дерматоглифики в рамки предметной области.

Примечания

1 См.: *Хомяков Э.Г.* Метод фазового анализа письменных объектов при проведении почерковедческих исследований: Дис. ... канд. юрид. наук. Ижевск, 2002.

2 См.: *Яковлева Е.В.* Криминалистическое исследование кратких записей, выполненных намеренно измененным почерком скорописным способом: Дис .... канд. юрид. наук. М.: Российской Федерации центр судебной экспертизы, 2006.

3 См.: *Орлова В.Ф.* Судебно-почерковедческая диагностика: Учеб. пособие для студентов вузов. М.: ЮНИТИ-ДАНА; Закон и право, 2006.

4 См.: *Шварц В.Б.* К проблеме врожденного и приобретенного в развитии двигательных способностей // Проблемы генетической психофизиологии. М., 1978; *Марютина Т.М.* О генотипической обусловленности вызванных потенциалов человека // Там же; *Сергиенко Л.П., Кореневич В.П.* Соотношение влияния наследственности и среды в процессе обучения движениям человека // Вопросы психологии. 1989. № 4.

5 *Кулик С.Д., Никонец Д.А., Ткаченко К.И.* Решение задач криминалистики при исследовании почерка кратких записей // Научная сессия МИФИ-2007. Сб. науч. трудов: В 17 т. Т. 12: Информатика и процессы управления. Компьютерные системы и технологии. М.: МИФИ, 2007. С. 24–25.

6 См.: *Гусакова С.М., Комаров А.С., Устинов В.В., Федорович В.Ю.* Применение ДСМ-метода к решению задач почерковедческой экспертизы // X национальная конференция по искусственному интеллекту «КИИ-2006», Обнинск, 25–28 сентября 2006: Труды конференции. Т. 1. М.: Физматлит, 2006; *Гусакова С.М., Комаров А.С.* Возможности применения ДСМ-метода для решения задач почерковедческой экспертизы // НТИ. Сер. 2. 2007. № 10.

7 *Финн В.К.* Правдоподобные рассуждения в интеллектуальных системах типа ДСМ // Итоги науки и техники. М., 1991. Т. 15. С. 54–98.

8 *Гусакова С.М., Комаров А.С., Устинов В.В., Федорович В.Ю.* Критерий достаточного основания как средство интеллектуального анализа криминалистических данных // Третья Международная конференция «Системный анализ и информационные технологии», Звенигород, 14–18 сентября 2009: Труды конференции. М., 2009. С. 173–175.

Abstracts

D.A. Larin

INFORMATION SECURITY IN ANCIENT RUSSIA

In this article process of information protection methods formation in our country and the first facts of their usage are considered. Main attention is given cryptographic methods of information protection, examples of Old Russian code numbers are resulted. Article covers time frames basically from IX century till the end of XIV century.

Keywords: hidden writing, cryptography, cipher, steganography.

V.V. Belov, A.V. Nekrakha

PATENT INFORMATION VALUE FOR INNOVATION DEVELOPMENT OF THE COUNTRY

Purpose: To analyze the problem of modernization national patent information system. On the basic of official statistics published by Word Intellectual Property Organization a relation between quality of system of patent information and level of national economy are established. The authors carry out the study information demand of users in patent documentation. Now patent information is the most effective source of knowledge about modern technology. From this point of view database of patent office of USA and database of European patent office present a great interest. The structure of said databases and main principles of information search in the databases are described. Special attention is devoted to database of Russian Patent Office. Unfortunately comparison between our national patent information system and the systems of developed countries is not in favor of national system. Taking into account this fact the authors make number proposals for modernization the system of patent information in our country.

Keywords: information, database, patent, modernization, patent office, patent search, economy, modern technology, information source.

M.I. Zabezhaylo

TOWARDS CHOICE OF ADEQUATE METHODOLOGY OF LARGE COMMERCIAL BANK'S BUSINESS TRANSFORMATION

Features of the IT Infrastructure modernization process for Russian commercial banks are discussed. The set of crucial for these projects success requirements is allocated. Variants for minimization of most important architectural and organizational design risks are offered.

Keywords: bank business IT infrastructure, IT Infrastructure modernization, architecture of bank information systems, the component-integration approach, the control and minimization of design risks.

T.A. Asmolov

TYPICAL INFORMATION SYSTEMS OF CULTURE INSTITUTIONS AND THEIR DEFENCE METHODS ANALYSIS

In the article the typical information systems applied in culture establishments are considered. Protection systems of the considered systems are analyzed, as well as principles according to which they are designed. The analysis of possible threats and risks helps to choose security measures, which should be carried out to reduce risk to comprehensible level. The author carries out the analysis of possible threats to information systems and of actions complex directed on their prevention. Classification of attacks to information systems and ways of their fulfilment is offered. In the conclusion the general strategy connected with decision-making in the conditions of information systems influence risk are allocated.

Keywords: information systems, information protection methods, information security threats, information systems protection methods, culture establishments, information resources, information system vulnerability.

S.V. Kudinov

ABOUT PRESENT TROUBLES IN INFORMATION SECURITY ECONOMICS

Modern problems of information security economics are stated. The special attention is given to practical aspect of the researches results application in the field of safety economy in the organisations. In par-

ticular, a method of projects estimation on the basis of weighted factor scoring models is analyzed. On the basis of the literature analysis the method features and its enhancement way for implementation are shown.

Keywords: information security economy, the investment analysis, weighted factor scoring models, information security problems, information security estimation.

A.N. Priezzhaya

ANALYSIS OF REGULATORY AND PROCEDURAL DOCUMENTS ON PERSONAL DATA PROTECTION

While providing information security personal data (PD) operators frequently face with a need of FSTEK and FSB Russia requirements observation, that represents not trivial problem. In given article the method for threats model and intruder possibilities design is offered, corresponding standard regulators. The method developed by the author is based on revealing of key concepts and constructing on their basis the consistent terminology and model.

Keywords: personal data, standard documents, threats model, intruder model, the requirements.

E.R. Beybutov

ESTIMATION PROCEDURE OF PERSONAL DATA OPERATORS INFORMATION SECURITY CORRESPONDENCE TO LEGISLATIVE REQUIREMENTS ON PERSONAL DATA PROTECTION

The federal law from July, 27th, 2006 № 152-FZ «About the personal data» has assigned on organizations, carrying out the personal data processing, the responsibility for safety requirements observance for any information concerning the physical person defined on the basis of such information (the subject of the personal data). The regulatory legal acts accepted on the basis of the specified Federal law have added and have specified information security requirements, having created theory-practical basis for works on the personal data protection. In the present article the technique, allowing to solve an actual problem of requirements observation estimation in the field of personal data protection is offered. The author carries out the analysis of regulatory legal acts, ordering of safety requirements on three levels is offered:

Management, organizational, software-technical, with their further decomposition, allowing to perform a quantitative and qualitative expert estimation of operators conformity to these requirements.

Keywords: the personal data legislation, the personal data protection, information security requirements, criteria of meeting the requirements.

I.V. Gaynanova

GAME THEORY APPLICATION FOR COVERT CHANNELS WITH ACTIVE ADVERSARY ANALYSIS

The purpose of the given work is research of active adversary behavior model provided that he does not analyze incoming messages in the data channel but only detains some of them with certain probability q. With usage of game theory methods some optimum likelihood values for various agent and auditor behavior are found as well as the covert channel capacity estimation in the set optimality conditions is received.

Keywords: covert channel, information security, active adversary, game theory.

E.I. Poznyakova

OPERATIONAL NEEDS ANALYSIS FOR QUALITY OF SERVICES REQUIREMENTS ESTABLISHMENT

The probability analysis method of operational needs was applied till now only for quality of services requirements definition for railways. In given article this method transparency for the other problems solving is considered, in particular for a business continuity factors estimation.

Keywords: quality of services, security requirements, business continuity, functional safety.

D.K. Skachek

BUILDING THE CONSISTENT CRITERIA SEQUENCE ON CONDITION OF MEASURE FUNCTION INCOMPLIANCE

Existence of the consistent criteria sequence point at possibility of anomalous system behavior revealing. We assume that the system condition is described by finite-dimensional topological space, on each of which the probability measure is set. The purpose of the given work –

to provide an example of the probability measures set, not satisfying to consistency constraints, and such that it is possible to construct consistent criteria sequence for abnormal behavior definition.

Keywords: inconsistent measures, consistent criteria sequence, hypothesis testing.

A.V. Gusev

RANDOM GRAPH METHOD FOR STEGANOGRAPHIC ANALYSIS OF CONTAINERS REPRESENTED AS GAUSSIAN PROCESSES

This work researches the operability of steganographic scheme based on Gaussian processes with computer modeling method. The possibility of steganography detection using random graph method is researched.

Keywords: steganography, computer modeling, gaussian processes, random graph.

S.Y. Melnikov

THE BOOLEAN FUNCTION STATISTICAL PROPERTIES CHARACTERIZED BY POLYGONS

For the given Boolean function f of the n arguments we introduce the locus of points, which coordinates are the relative frequencies limits of the symbols “one” in growing input and output sequences of the binary shift register of the span n and the output function f. We prove this locus of points to be a convex polygon. The article deals with the way how the algebraic properties of the Boolean function are reflecting in the geometrical properties of the polygon.

Keywords: de Bruijn graph, shift register, Boolean function, convex polygon.

A.A. Lipatiev

DIMENSIONALITY REDUCTION AND CLASSIFICATION

In this article the dimensionality reduction by a method of the main components and clusterization by a method of K-averages are considered. For the given combination of methods the principal pos-

sibility of clusterization by a method of K-averages for the data, which are the result of source data set dimensionality reduction, is proved.

Keywords: K-averages method, main components method, multivariance analysis.

Y.V. Kozlova

GENERATOR OF TEST DATA FOR DIFFERENT JSM-STRATEGIES

In this article a program that generates test data for demonstration particularities of different JSM-strategies is described. There are examined the examples of the experimental JSM-system usage with data generated by this program.

Keywords: JSM-method, JSM-system, variants of JSM-method, testing, test data.

M.A. Mikheyenkova

ABOUT FORMALIZED HEURISTIC SCHEME OF QUALITATIVE SOCIOLOGICAL DATA ANALYSIS

Two approaches to the formalizations of heuristic scheme "similarity – analogy – abduction" are proposed in the paper. The scheme is used in qualitative sociological data analysis, which is automatic generation of dependencies on the base of facts. The first approach is based on Boolean algebra for similarity implementation. The second is realized in Intelligent Systems using JSM-reasoning which represents the class of cognitive plausible reasoning.

Keywords: formalized qualitative analysis, JSM-method, automatic generation of hypotheses, intelligent system, plausible reasoning, boolean algebra.

Y.V. Kozlova

ABOUT OBJECT MODEL OF THE REQUIREMENTS' SYSTEM FOR THE PROGRAM "TestJSM!"

In this article a modified object model of the requirements' system for the program "TestJSM!" is described. It generates test data for demonstration of particularities of different JSM-strategies.

Keywords: JSM-method, JSM-system, variants of JSM-method, testing, test data, generator of test examples, object model, requirements' system for test examples.

A.N. Priezzhaya

COMPUTER-AIDED ENGINEERING OF PROTECTED INFORMATION SYSTEM

Design process of information system in the protected execution requires labour-consuming analysis of protected object and considerable quantity of documents. In given article is offered the technology based on automated transformation of UML-models which allow to work simultaneously with some representations of the object, including the text description, and to automatically build threats amd intruder model on their basis, and also protection system model. In given article protected information system design is considered as case study of the distributed database.

Keywords: UML-model, the personal data, model transformation, object model, threats model, the documentation, system model in the protected execution.

Y.K. Sergeev

ANALYSIS OF SOME MECHANISMS FOR VIRTUAL MACHINES MEMORY MANAGEMENT

Malicious software evolution, expressed in growth of possibilities of these programs on realisation of unauthorized operations in information systems, conducts behind itself protection frames development. Every year viruses, net worms and Trojan programs all become more intellectual, so all new means are applied to struggle against them, new approaches are searched. While knowing the established information protection frames (IPF) on an attacked host, the malefactor can always develop new technology for malware concealment from the set security tools. The author sets as the purpose to design such system in which there is a possibility to provide IPF invisibility for the malefactor and/or malware within the OS in which intrusion is performed, by removing IPF outside its frameworks. For such architecture realisation it is offered to lean against virtualization technologies with hypervisor application. In a kind of opened source codes (Open Source) for this purpose achievement it is used Xen hypervisor. To investigate applica-

tion possibility of the given technology, it is necessary to consider the mechanisms, allowing to isolate virtual machines from each other. One of key mechanisms is operative memory management on virtual machines.

Keywords: hypervisor, operative memory management mechanisms, information protection tools, virtual machines isolation.

M.V. Levykin

SAFETY ANALYSIS OF STANDARD MECHANISM FOR REGISTER ACCESS CONTROL IN WINDOWS XP KERNEL

The system register plays a key role in configuration and management of OS Windows. This is the store of system and user parameters. The register is not static set of the data stored on a hard disk. It represents a set of various structures which are stored in computer memory and are supported by a kernel and executive system. The register is used during: system, Explorer and other components of OS loading, installation of applications and drivers, applications and drivers launch etc.

Now it is difficult to underestimate a role of the system register in questions of OS Windows as functioning practically all contemporary means of concealment, anyhow, is connected with the system register.

Keywords: system register, mechanism of register call back, executive system, the dispatcher of system services, the dispatcher of configuration, OS Windows kernel, the driver.

M.A. Borisov, I.V. Zavodtsev

TOOLS OF COMPUTER SYSTEMS VULNERABILITIES EVALUATION

In the article the concept of vulnerabilities estimation system is considered and the generalised approach to revealed vulnerabilities estimation is offered, considering world experience in this direction and using Russian experts best practice.

Keywords: automated system, information security, vulnerabilities estimation system, risks prioritizing, base metrics, contextual metrics, time metrics.

I.V. Shidlovsky-Moskvin

ANALYSIS OF PROTECTED SANDBOX BASED ON FULL VIRTUALISATION TECHNOLOGY, CASE STUDY OF VMWARE WORKSTATION

The virtualization technology has appeared in the sixties 20 century. However it has received the greatest development last years, first of all, because of hardware capacity, simplicity and its convenience growth, and the main thing is that new functions inaccessible earlier have appeared, such as simultaneous operation of several operating systems (OS) on one hardware platform, and, migration of working OS in real time mode between hardware platforms etc.

Now one of key virtualization functions is its usage as the safety mechanism. Namely – creation of sandbox (separate program environment) (SPE) based on virtualization technology.

However, many experts overestimate virtual machines security, and, hence, the sandboxes constructed on their basis.

Keywords: virtualization, virtualization types, sandbox, WinDbg, VMware Workstation, Smartline DeviceLock, SSDT, interception of the system services scheduling table.

A.E. Satunina, A.S. Sysoev

APPROACH TO SECURITY DESIGN IN SERVICE ORIENTED ARCHITECTURES

Article is devoted to the safety problems of corporate information systems (CIS) based on service oriented architecture (SOA). Approaches to safety, as separate kind of IT Service of the CIS, caused by SOA information systems features are considered. Processes of information security maintenance, demanding the system analysis and estimations of interrelations with other IT Services of CIS at all stages of life cycle are listed and characterized.

Keywords: corporate information systems security, service oriented architecture, IT Services, security in SOA-systems, infrastructure safe management, security of interrelations between IT Services, security management.

M.A. Borisov

TOWARDS INFORMATION SECURITY SYSTEM MODELING IN CONDITIONS OF INFORMATION WARFARE

In article the concept of information security is considered and the variant of base model for information protection system estimation in the conditions of the information antagonism is offered, operating on the basis of the information risks analysis and calculation of financial and economic indicators.

Keywords: information security, information warfare, information protection system, information value.

A.S. Komarov

INTELLECTUAL DATA ANALYSIS IN GRAPHOLOGY: SOFTWARE IMPLEMENTATION

This article describes software product for intelligence handwriting data analysis which meant for handwriting experiments and handwriting expertise support. Intelligence data analysis is based on JSM-method for automatic generation of hypotheses. This software product makes it possible to several researchers work with one system. The article specifies problem which developer solve now and plan to solve in the near future.

Keywords: intelligence data analysis, JSM-method, handwriting expertise, graphology, web application, multitier architecture.

Сведения об авторах

Асмолов Тимофей Александрович – директор по образовательным проектам ООО «Нота», tasmolov@yandex.ru.

Бейбутов Эльман Рафикович – аспирант кафедры компьютерной безопасности ИИНиТБ РГГУ, flash-best@mail.ru.

Белов Владимир Владимирович – кандидат технических наук, доцент кафедры организационно-правовой защиты информации ИИНиТБ РГГУ, zvvbelov@mtu-net.ru.

Борисов Михаил Анатольевич – доцент кафедры специальных дисциплин ВС РФ факультета военного обучения МГУ им. М.В. Ломоносова, bma_mv@rambler.ru.

Гайнанова Ирина Валерьевна – ведущий специалист ОАО «АТС», irina.gain@gmail.com.

Гусев Александр Витальевич – преподаватель кафедры компьютерной безопасности ИИНиТБ РГГУ, alexander_gusev@inbox.ru.

Забежайло Михаил Иванович – кандидат физико-математических наук, zmivan@gmail.com.

Заводцев Илья Валентинович – кандидат технических наук, доцент кафедры защиты информации в автоматизированных системах Краснодарского высшего военного училища (военного института) им. С.М. Штеменко, uilus@rambler.ru.

Козлова Юлия Владимировна – лингвист, компания «Наносемантика», juliaz-2001@yandex.ru.

Комаров Алексей Сергеевич – аспирант ВИНИТИ РАН, alexskomarov@gmail.com.

Кудинов Станислав Владимирович – аспирант кафедры компьютерной безопасности ИИНиТБ РГГУ, koudinov@mail.ru.

Ларин Дмитрий Александрович – кандидат технических наук, доцент ИКСИ, greattzar@yandex.ru.

Левыкин Михаил Владимирович – аспирант кафедры компьютерной безопасности ИИНиТБ РГГУ, de_shiko@yahoo.com.

Липатьев Александр Андреевич – технический специалист «СтатСофт Раша», workingtoday@mail.ru.

Мельников Сергей Юрьевич – кандидат физико-математических наук, специалист ООО «Стэл – Компьютерные Системы», melnikov@stel.ru.

Михеенкова Мария Анатольевна – кандидат технических наук, доцент УНЦ «Проблемы и методы интеллектуального анализа данных» отделения интеллектуальных систем в гуманитарной сфере, ma_mikh@bk.ru.

Некраха Андрей Вячеславович – кандидат технических наук, декан факультета защиты информации ИИНТБ РГГУ, rggu.abc@mail.ru.

Познякова Екатерина Игоревна – аспирантка кафедры компьютерной безопасности ИИНиТБ РГГУ, e.poznyakova@gmail.com.

Приезжая Алина Николаевна – аспирантка кафедры компьютерной безопасности ИИНиТБ РГГУ, alina_pr@list.ru.

Сатунина Анна Евгеньевна – кандидат экономических наук, ведущий научный сотрудник, декан факультета информатики ИИНиТБ РГГУ, aesat@mail.ru.

Сергеев Юрий Константинович – аспирант кафедры компьютерной безопасности ИИНиТБ РГГУ, ysergeev@gmail.com.

Скачек Дарья Константиновна – специалист отдела информационных технологий дирекции целевых программ ЗАО «РНТ», sol_aurora@mail.ru.

Сысоев Александр Сергеевич – аспирант факультета информатики ИИНиТБ РГГУ, zt0@mail.ru.

Шидловский-Москвин Иван Витальевич – аспирант кафедры компьютерной безопасности ИИНиТБ РГГУ, ivashi@mail.ru.

General data about the authors

Asmolov Timofey A. – head of educational projects in LLC "NOTA", tasmolov@yandex.ru.

Beybutov Elman R. – postgraduate of computer security department of IISaST of RSUH, flash-best@mail.ru.

Belov Vladimir V. – candidate of engineering science, associate professor of organizational-legal information security of IISaST of RSUH, zvvbelov@mtu-net.ru.

Borisov Michael V. – associate professor of special disciplines department of Military education faculty, MSU n.a. Lomonosov, bma_mv@rambler.ru.

Gaynanova Irina V. – lead specialist, OJSC "ATS", irina.gain@gmail.com.

Gusev Alexander V. – professor of computer security department of IISaST of RSUH, alexander_gusev@inbox.ru.

Zabezhaylo Michael I. – candidate of physical and mathematical sciences, zmivan@gmail.com.

Zavodtsev Ilya V. – candidate of engineering science, associate professor in department of computer systems information security, Krasnodar Higher Military School n. a. S.M. Shtemenko, uilus@rambler.ru.

Kozlova Yulia V. – linguist, Nanosemantics, juliaz-2001@yandex.ru.

Komarov Alexey S. – postgraduate of All-Russian Institute for Scientific and Technical Information (VINITI RAN), alexskomarov@gmail.com.

Kudinov Stanislav V. – postgraduate of computer security department in IISaST of RSUH, koudinov@mail.ru.

Larin Dmitry A. – candidate of engineering science, associate professor of IKSI, greattzar@yandex.ru.

Levykin Michael V. – postgraduate of computer security department in IISaST of RSUH, de_shiko@yahoo.com.

Lipatiev Alexander A. – technical specialist "StatSoft Russia", workingtoday@mail.ru.

Melnikov Sergey Y. – candidate of physical and mathematical sciences, specialist LLC "Stel – Computer Systems", melnikov@stel.ru.

Mikheyenkova Maria A. – Ph.D., Educational Research Center "Problems and Methods of Intelligent Data Analysis", Department of Intelligent Systems for the Humanities, associate professor, ma_mikh@bk.ru.

Nekrakha Andrey V. – candidate of engineering science, dean of information security faculty in IISaST of RSUH, rggu.abc@mail.ru.

Poznyakova Ekaterina I. – postgraduate of computer security department in IISaST of RSUH, e.poznyakova@gmail.com.

Priezzhaya Alina N. – postgraduate of computer security department in IISaST of RSUH, alina_pr@list.ru.

Satunina Anna E. – candidate of economics, leading research associate, dean of computer science faculty in IISaST of RSUH, aesat@mail.ru.

Sergeev Yuri K. – postgraduate of computer security department in IISaST of RSUH, ysergeev@gmail.com.

Skachek Daria K. – specialist of information technologies department in target programs front office of CJSC "RNT", sol_aurora@mail.ru.

Sysoev Alexander S. – postgraduate of computer science faculty in IISaST of RSUH, zt0@mail.ru.

Shidlovsky-Moskvin Ivan V. – postgraduate of computer security department in IISaST of RSUH, ivashi@mail.ru.

Заведующая редакцией *Е.Е. Жигарина*
Корректор *Л.П. Бурцева*
Художник номера *В.Н. Хотеев*
Компьютерная верстка *Е.Б. Рагузина*

Подписано в печать 02.07.2010.
Формат 60×90 $^{1}/_{16}$.
Усл. печ. л. 19,75. Уч.-изд. л. 20,0.
Тираж 1050 экз. Заказ № 141.

Издательский центр
Российского государственного
гуманитарного университета
125993, Москва, Миусская пл., 6
www.rggu.ru
www.knigirggu.ru