МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт
Компьютерных
Технологий и
Информационной
Безопасности

СОВРЕМЕННЫЕ МЕТОДЫ, СРЕДСТВА И ТЕХНОЛОГИИ ЗАЩИТЫ ИНФОРМАЦИИ – 2024

Сборник трудов

XV Международной научно-практической
конференции имени Олега Борисовича Макаревича

Посвящается 90-летию профессора
Макаревича Олега Борисовича

Таганрог, 11–15 сентября 2024

Ростов-на-Дону – Таганрог
Издательство Южного федерального университета
2024

УДК 004.56(063)
ББК 16.8 я431
С56

С56 **Современные методы, средства и технологии защиты информации – 2024** [Электронный ресурс] : сборник трудов XV Международной научно-практической конференции имени Олега Борисовича Макаревича (Таганрог, 11–15 сентября 2024 г.) ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2024. – Электрон. текстовые дан. (1 файл: 4,89 Мб). – 1 электрон. опт. диск (CD-R). – Системные требования: процессор с тактовой частотой 1,5 ГГц и выше, 2 Гб оперативной памяти, Windows 7 SP1/8, 8.1/10/11 (64-разрядная версия), Acrobat Reader, привод DVD-ROM. – Загл. с экрана. – 263 с.
ISBN 978-5-9275-4813-2

В сборник трудов XV Международной научно-практической конференции имени Олега Борисовича Макаревича вошли статьи по следующим направлениям: «Методы и системы информационной безопасности»; «Вопросы информационной безопасности автоматизированных систем и систем связи»; «Методы сетевой безопасности»; «Безопасность распределенных систем и телекоммуникаций»; «Безопасность критических информационных инфраструктур»; «Теоретические и практические аспекты криптографии»; «Безопасность киберфизических систем и БПЛА»; «Безопасность программного обеспечения»; «Правовые основы и защита государственной, коммерческой тайны и интеллектуальной собственности»; «Новые нормативные правовые и методические документы, регламентирующие деятельность по защите информации и объектов»; «Подготовка специалистов в области информационной безопасности»; «Безопасность систем Искусственного Интеллекта».

Материалы публикуются в авторской редакции.

ISBN 978-5-9275-4813-2

УДК 004.56(063)
ББК 16.8 я431

ПРОГРАММНЫЙ КОМИТЕТ

Председатель

Веселов Г.Е. – д.т.н., доцент, директор Института компьютерных технологий и информационной безопасности (ИКТИБ) Южного федерального университета (ЮФУ), Россия.

Зам. председателя

Бабенко Л.К. – д.т.н., профессор, профессор кафедры «Безопасность информационных технологий» (БИТ) им. О.Б. Макаревича ИКТИБ ЮФУ, Россия.

ЧЛЕНЫ ПРОГРАММНОГО КОМИТЕТА

Atilla Elci – Professor, PhD, Hasan Kalyoncu University, Türkiye;

Luis Ramiro Piñeiro Díaz – Professor, PhD, Cryptography Institute of the Faculty of Mathematics and Computing of the University of Havana, Cuba;

Maxim Anikeev – Associate Professor, PhD, Fraunhofer SIT | ATHENE, researcher, Germany;

Miguel Katrib Mora – Professor, PhD, Cryptography Institute of the Faculty of Mathematics and Computing of the University of Havana, Cuba;

Mohd Helmy Abd Wahab – Professor, PhD, Universiti Tun Hussein Onn Malaysia;

Pradeep Kumar Singh – Professor, PhD, Central University of Jammu, India;

Абрамов Е.С. – к.т.н., доцент, заведующий кафедрой БИТ им. О.Б. Макаревича ИКТИБ ЮФУ, Россия;

Алгазы К.Т. – с.н.с., PhD, Институт информационных и вычислительных технологий комитета науки министерства науки и высшего образования Республики Казахстан (ИИВТ КН МНВО РК), Казахстан;

Белов Е.Б. – заместитель председателя ФУМО ВО ИБ, Россия

Ищукова Е.А. – к.т.н., доцент, доцент кафедры БИТ им. О.Б. Макаревича ИКТИБ ЮФУ, Россия;

Калмыков И.А. – д.т.н., профессор, профессор кафедры информационной безопасности автоматизированных систем (ИБАС) Северо-Кавказского федерального университета (СКФУ), Россия;

Капалова Н.А. – г.н.с. ИИВТ КН МНВО РК, Россия;

Климов С.М. д.т.н., профессор 4ый Центральный научно-исследовательский институт, Россия;

Конявский В.А. – д.т.н., заведующий кафедрой «Информационная безопасность» Московского физико-технического Института (МФТИ), Россия;

Котенко И.В. – д.т.н., профессор Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН), Россия;

Машкина И.В. – д.т.н., профессор, профессор кафедры «Вычислительная техника и защита информации» Уфимского университета науки и технологий, Россия;

Милославская Н.Г. – д.т.н. и PhD in Cyber Security (UK), доцент, профессор Национального исследовательского ядерного университета «МИФИ», Россия;

Марков А.С. – д.т.н., профессор, Президент группы компаний «Эшелон», профессор кафедры ИУ-8 «Информационная безопасность» МГТУ им Н.Э. Баумана, Россия;

Осипян В.О. – д.ф-м.н., доцент, профессор кафедры анализа данных и искусственного интеллекта Кубанского государственного университета, Россия;

Пересыпкин В.А. – д.т.н., действительный член Академии криптографии Российской Федерации, Россия;

Петренко В.И. – к.т.н., доцент, заведующий кафедрой организации и технологии защиты информации института цифрового развития, директор института цифрового развития СКФУ, Россия;

Спиридонов О.Б. – к.т.н., директор НКБ «МИУС» ЮФУ;

Тебуева Ф.Б. – д.ф-м.н., доцент, заведующая кафедрой компьютерной безопасности СКФУ, Россия;

Целых А.Н. – д.т.н., профессор, заведующий кафедрой «Информационно-аналитические системы безопасности» имени профессора Л. С. Берштейна, Россия;

Чефранов А.Г. – д.т.н., профессор, доцент кафедры вычислительной техники, Восточно-средиземноморский университет, Северный Кипр;

Шелупанов А.А. – д.т.н., профессор, президент Томского государственного университета систем управления и радиоэлектроники (ТУСУР), директор Института системной интеграции и безопасности, Россия

Язов Ю.К. – д.т.н., профессор, г.н.с. ГНИИ ПТЗИ ФСТЭК, Россия.

УДК 004.056.5; 004.052.42

И.М. Ажмухамедов, А.В. Хайтул
Россия, г. Астрахань, Астраханский государственный университет им. В.Н. Татищева

ДОСТОВЕРНОСТЬ КАК ПРОБЛЕМА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

В данной работе авторами приведено собственное определение понятия «достоверность». Описаны действия, способствующие искажению информации. Рассмотрен феномен субъективной оценки степени достоверности, влияющую на восприятие информации. Проанализирована взаимосвязь между достоверностью и классическими сервисами информационной безопасности: конфиденциальностью, целостностью, доступностью, аутентичностью и неотказуемостью. Обоснована необходимость включения достоверности информации в список классических сервисов, обеспечивающих информационную безопасность.

Ключевые слова: *информационная безопасность; достоверность информации; деструктивное влияние; сервисы информационной безопасности.*

In this work, the authors provide their own definition of the concept of «trustworthiness». The actions that contribute to the distortion of information are described. The phenomenon of subjective assessment of the degree of reliability, influencing the perception of information, is considered. The interrelation between do-reliability and classical services of information security: confidentiality, integrity, availability, authenticity and unreliability are analyzed. The necessity of including information reliability in the list of classical services providing information security is substantiated.

Keywords: *information security; information reliability; destructive influence; information security services.*

Введение

Обмен информацией является неотъемлемой частью жизни человеческого общества, и с развитием сети Интернет, он становится все более интенсивным. Этот глобальный тренд повышает значимость вопросов информационной безопасности, поскольку с ростом онлайн-активности возрастает и степень подверженности различным угрозам.

Обеспечение информационной безопасности (ИБ) обычно предполагает рассмотрения таких классических сервисов как конфиденциальность, целостность, доступность, аутентичность и неотказуемость. Однако эти аспекты касаются главным образом защиты собственной информации.

В то же время, с увеличением распространения деструктивной информации, возрастает энтропия в социальной системе, появляются риски искажения действительности, что может привести к разрушительным и часто необратимым последствиям.

При этом необходимо отметить, что существенная часть деструктивной информации является по своей сути недостоверной.

Исходя из вышеизложенного, представляется целесообразным расширить перечень сервисов, рассматриваемых в рамках ИБ, включив в него понятие «достоверность», поскольку оно является одним из важнейших имманентных (внутренне присущих объекту) свойств информации и может оказывать существенное влияние на безопасность как отдельных пользователей, так и общества в целом [1].

Определение понятия достоверность

Различные источники предлагают разные определения понятия «достоверность» [2–9].

Несмотря на разнообразие, во всех определениях имеются общие, инвариантные черты, которые подчеркивают для достоверности необходимость наличия таких свойств как высокая степень надежности, правдоподобия информации, знаний или данных.

Помимо этого, во всех определениях указывается, что достоверность связана с уверенностью в том, что информация является точной и может быть доказана, исключая сомнения в ее ненадежности.

Исходя из вышеизложенного, в работе [1] было предложено следующее определение: достоверной считается информация, точно и объективно отражающая события, произошедшие в конкретном (соответствующем действительности) месте и в указанное время.

Согласно данному определению важно, чтобы сведения о событии, месте и времени были достоверны в совокупности.

Действия, приводящие к недостоверности информации

Искажение сути события или факта. Природа данного действия может носить как объективный, так и субъективный характер. Объективное (непреднамеренное) искажение обычно возникает на начальной стадии освещения того или иного события или факта, когда высока степень неопределенности, неизвестны многие обстоятельства произошедшего.

Кроме этого, может иметь место субъективное (преднамеренное) искажение фактов. Источник, распространяющий информацию в сети Интернет, может сознательно вводить в заблуждение пользователей различными способами: выборочное представление фактов, придание им определенной эмоциональной окраски, интерпретация событий в «выгодном» свете, некорректное цитирование источников и т.д.

Все вышеуказанные способы могут использоваться как в отдельности, так и в совокупности для создания иллюзии достоверности, в то время как сама информация может быть ложной или искаженной.

Недостоверное указание места. Такое действие подрывает доверие к источнику, особенно если эта информация является важной для понимания контекста.

Недостоверное указание времени. Доверие к информации может быть подорвано также некорректным указанием времени, когда произошло то или иное событие. Такие ошибки в указании времени могут возникать в результате влияния человеческого фактора, технических сбоев, задержек при получении сведений и т.п.

Субъективная оценка достоверности информации

Люди принимают решения, основываясь на собственной (субъективной) уверенности в достоверности полученной информации. То есть их поступки основаны на том, на сколько информация кажется им надёжной. При этом одно и то же утверждение, которое кажется одному человеку достоверным, другому может показаться сомнительным.

Субъективная оценка степени достоверности информации зависит от ряда факторов:

1. Люди могут иметь предвзятое мнение, которое влияет на восприятие ими информации.

2. Эмоциональное состояние также оказывает влияние на восприятие и оценку информации. Яркие эмоции, такие как, например, страх или гнев сильно мешают объективному анализу.

3. Люди в гораздо большей степени доверяют тому, что соответствует их личному опыту, и сомневаются в информации, которая ему противоречит.

4. Мнение, распространённое в социальной среде окружающей человека, также оказывает сильное воздействие на процесс восприятия им информации.

5. Информация, которая соответствует этическим или моральным убеждениям человека, обычно воспринимается им как более достоверная.

6. Люди склонны переносить доверие к источнику информации на саму информацию, полученную из этого источника.

7. Имеет значение и форма представления информации. Так, например, информация, представленная в форме научного доклада, в большинстве случаев считается более достоверной, чем информация, представленная в виде частного мнения, изложенного в социальных сетях.

Исходя из вышеизложенного, можно утверждать, что степень субъективной оценки уровня достоверности зависит от следующего множества факторов, указанных в формуле 1:

$$<B; F; E; P; C; S; R>, \quad (1)$$

где B – Bias (предвзятость), F – Feelings (эмоции), E – Experience (опыт), P – Perception (восприятие), C – Conviction (убеждение), S – Source (источники), R – Representation (форма представления).

Таким образом, информация, циркулирующая в цифровой среде, может быть как полностью, так и частично недостоверной, что с учётом субъективных особенностей пользователей цифровых сервисов часто приводит к принятию ими неверных решений и может оказывать на них деструктивное воздействие.

Достоверность и классические сервисы информационной безопасности

Рассмотрим взаимосвязь достоверности с основными (классическими) сервисами информационной безопасности, которые были перечислены ранее.

Конфиденциальность представляет собой такое свойство информации, которое обеспечивает недоступность сведений для нелегитимного пользователя.

Требования к достоверности и конфиденциальности часто противоречат друг другу, поскольку открытое, неограниченное распространение информации в значительной мере способствует выявлению факта её недостоверности. В случае с конфиденциальной информацией такого рода возможности весьма ограничены.

Целостность является свойством информации, означающим, что данные остаются неискажёнными в течение всего их жизненного цикла, то есть они не подвергаются несанкционированным изменениям, искажениям.

Взаимосвязь между целостностью и достоверностью заключается в том, что нарушение целостности ведёт к тому, что информация часто становится недостоверной.

Доступность – это свойство информации, которое обеспечивает гарантию получения данных легитимным пользователям в любое время.

Потеря доступности к объективной информации способствует увеличению спроса на альтернативные информационные ресурсы. Однако они могут быть менее качественными и информация, полученная из такого рода источников, часто может оказаться недостоверной, что в последствии повышает вероятность появления и распространения фейковой информации. Следовательно, обеспечение доступности к объективной информации играет ключевую роль в сдерживании распространения фейковых (недостоверных) сведений.

Кроме этого, доступность помогает пользователям в проверке достоверности данных, так как при открытом доступе к информации люди могут свободно убедиться в том, что событие реально случилось.

Аутентичность и неотказуемость взаимодополняющие свойства, которые направлены на обеспечение доверия к происхождению и целостности информации. Аутентичность помогает убедиться в

идентификации субъекта, распространяющего информацию, и проверяет не было ли подделано или изменено авторство в процессе распространения данных. Неотказуемость обеспечивает невозможность отказа от передачи или приема информации субъектами информационных отношений.

Нарушение этих сервисов так же, как и в случае нарушения сервиса «доступность» создаёт благоприятные условия и провоцирует распространение недостоверной информации, поскольку у создателей такого рода информации возникает чувство безнаказанности (в большинстве своём вполне обоснованное).

Исходя из вышеизложенного, рассмотренные взаимосвязи можно представить в виде следующей схемы.

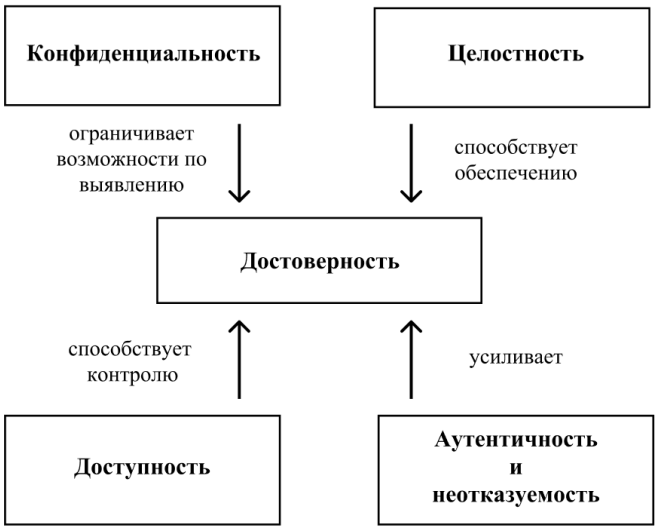


Рис. 1. Взаимосвязь достоверности и классических сервисов информационной безопасности

Таким образом, можно сделать вывод о том, что существует тесная взаимосвязь между достоверностью и классическими сервисами ИБ.

Заключение

Обеспечение достоверности информации в цифровую эпоху становится всё более актуальной задачей, объединяющей в себе различные аспекты в области информационной безопасности.

При этом усилия по обеспечению достоверности данных в цифровой среде должны опираться как на технические и технологические аспекты, так и на развитие культуры цифровой грамотности и осмысленного подхода в использовании и распространении информации в сети Интернет.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Ажмухамедов И.М., Хайтул А.В.* Достоверность как сервис информационной безопасности в цифровой среде // Прикаспийский журнал: управление и высокие технологии. – 2023. – № 4 (64). – С. 26-35. – EDN FCZLUO.
2. Большой энциклопедический словарь // Wikipedia: сайт. – URL: https://ru.wikipedia.org/wiki/Большой_энциклопедический_словарь (дата обращения: 21.07.2024).
3. *Ивин А.А., Никифоров А.Л.* Словарь по логике. – М.: Гуманитарный издательский центр ВЛАДОС, 1997. – 384 с.
4. Философский энциклопедический словарь // Словари онлайн: сайт. – URL: https://rus-philosophy-enc.slovaronline.com/ (дата обращения: 21.07.2024).
5. Сборник энциклопедий и словарей русского языка // Значение слова Достоверность в энциклопедическом словаре: сайт. – URL: https://diclist.ru/slovar/enciklopedicheskiy/d/1-dostovernost.html (дата обращения: 21.07.2024).
6. Сборник энциклопедий и словарей русского языка // Значение слова Достоверность в словаре логики: сайт. – URL: https://diclist.ru/slovar/logiki/d/dostovernost.html (дата обращения: 21.07.2024).
7. Сборник энциклопедий и словарей русского языка // Значение слова Достоверность в философском словаре: сайт. – URL: https://diclist.ru/slovar/logiki/d/dostovernost.html (дата обращения: 21.07.2024).
8. Большой толковый социологический словарь // gufo.me: сайт. – URL: https://gufo.me/dict/social_dict?page=5&letter=д (дата обращения: 21.07.2024).
9. Большой толковый социологический словарь // Словари онлайн: сайт. – URL: https://rus-philosophy-enc.slovaronline.com/ (дата обращения: 21.07.2024).

УДК 004.056.55

Л.К. Бабенко, В.С. Стародубцев

Россия, г. Таганрог, Южный федеральный университет

ОЦЕНКА СЛОЖНОСТИ ВЫПОЛНЕНИЯ ОПЕРАЦИЙ ШИФРОВАНИЯ И РАСШИФРОВАНИЯ СИММЕТРИЧНОЙ ВЕРОЯТНОСТНОЙ ГОМОМОРФНОЙ КРИПТОСИСТЕМОЙ ДОМИНГО-ФЕРРЕРА

В данной статье оценивается сложность выполнения операций шифрования и расшифрования симметричной вероятностной гомоморфной криптосистемой Доминго-Феррера, основанной на задаче факторизации чисел. Для операций шифрования и расшифрования приводятся формулы расчёта количества выполняемых базовых математических операций, а также графики, отражающие зависимости количества операций от выбранных параметров криптосистемы. Результатом исследования являются выявленные наиболее трудоёмкие этапы шифрования и расшифрования криптосистемой Доминго-Феррера, подтвержденные экспериментально. Показано, что для повышения эффективности операций шифрования и расшифрования требуется оптимизация.

Ключевые слова: *Информационная безопасность; конфиденциальная информация; гомоморфное шифрование; криптосистема Доминго-Феррера; оценка сложности алгоритма шифрования.*

This article estimates the complexity of encryption and decryption operations by the symmetric probabilistic homomorphic Domingo-Ferrer cryptosystem based on the number factorization problem. For encryption and decryption operations, formulas for calculating the number of basic mathematical operations performed are provided, as well as graphs reflecting the dependence of the number of operations on the selected parameters of the cryptosystem. The result of the study is the identification of the most labor-intensive stages of encryption and decryption by the Domingo-Ferrer cryptosystem, confirmed experimentally. It is shown that optimization is required to improve the efficiency of encryption and decryption operations.

Keywords: *information security; confidential information; homomorphic encryption; Domingo-Ferrer cryptosystem; evaluation of the complexity of the encryption algorithm.*

Описание криптосистемы Доминго-Феррера

Данная криптосистема была разработана в 1996 году и обеспечивает выполнение гомоморфных операций сложения, вычитания и умножения [1–3]. Система является симметричной, из чего следует, что как для процесса шифрования, так и для расшифрования применяется один ключ [4]. Краткое описание операций, выполняемых в криптосистеме Доминго-Феррера, приведено на рис. 1. В сравнении с криптосистемами, разработанными на основе методов, представленных Джентри [5–8], их реализация требует значительно меньших вычислительных ресурсов. Однако необходимо подчеркнуть, что для использования таких криптосистем на практике требуется их всестороннее исследование.

Рис. 1. Операции шифра Доминго-Феррера

Количество выполняемых операций выбора случайного элемента при шифровании

На начальном этапе шифрования в криптосистеме Доминго-Феррера формируется набор случайных чисел $a_2 \ \ldots \ a_d$. Числа $a_2, \ldots, a_{d-1}$ заполняются по формуле (1).

$$a_i \xleftarrow{\$} Z_n, \tag{1}$$

где i – индекс в диапазоне [2; d – 1]; Z_n – кольцо по модулю n.

Число a_d также получает случайное значение из Z_n, однако на него накладывается дополнительное ограничение – выбранное число должно быть ненулевым. Данное условие отражено в формуле (2).

$$a_d \overset{\$}{\leftarrow} Z_n \setminus \{0\}. \quad (2)$$

Из приведенных формул (1) и (2) следует, что для формирования набора случайных чисел $a_2 \ \dots \ a_d$ требуется выполнить $d-1$ операций выбора случайного элемента.

Количество выполняемых операций сложения и вычитания при шифровании

Числу a_1 присваивается значение в соответствии с формулой (3).

$$a_1 = m - \left(\sum_{i=2}^{d} a_i\right) mod\ n, \quad (3)$$

где m – блок шифруемого сообщения, n – модуль, определенный в параметрах криптосистемы.

Из формулы (3) следует, что для вычисления a_1 из значения открытого текста m вычитается сумма всех чисел из набора случайных чисел $a_2 \ \dots \ a_d$. Для сложения $d-1$ слагаемых требуется выполнить $d-2$ операций сложения. Таким образом, в криптосистеме Доминго-Феррера в процессе шифрования выполняется $d-2$ операций сложения и одна операция вычитания.

Количество выполняемых операций умножения при шифровании

В криптосистеме Доминго-Феррера при шифровании на заключительном этапе при формировании пары полиномов $(\pi(x), \rho(x))$ шифртекста выполняются операции умножения, как показано в формулах (4) и (5).

$$\pi(x) = \left(a_d \cdot r_p^d \, x^d + \cdots + a_1 \cdot r_p \, x\right) mod\ p. \quad (4)$$

$$\rho(x) = \left(a_d \cdot r_q^d \, x^d + \cdots + a_1 \cdot r_q \, x\right) mod\ q. \quad (5)$$

Как видно из формул (4) и (5), для формирования коэффициентов полинома шифртекста для каждого из них выполняется количество умножений, равное степени полинома, перед которой установ-

лен данный коэффициент [9, 10]. Таким образом, в криптосистеме Доминго-Феррера для формирования пары полиномов шифртекста количество умножений $MulCount$ определяется по формуле (6).

$$MulCount = 2 \cdot \sum_{i=1}^{d} i, \tag{6}$$

где d – степень полинома представления шифртекста.

График зависимости количества операций умножения от степени полинома представления шифртекста приведен на рис. 2.

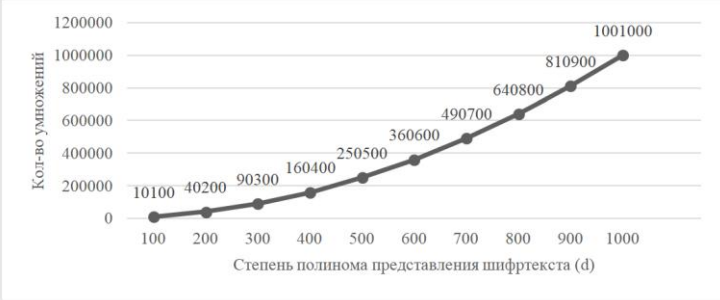


Рис. 2. Зависимость количества операций умножения от степени полинома представления шифртекста при шифровании

Количество выполняемых операций получения остатка от деления при шифровании

В процессе шифрования криптосистемой Доминго-Феррера операция получения остатка от деления используется при вычислении числа a_1 (формула (3)) и при формировании пары полиномов $(\pi(x), \rho(x))$ шифртекста (формулы (4) и (5)). При вычислении числа a_1 выполняется одна операция $mod\ n$, а при формировании пары полиномов $(\pi(x), \rho(x))$ шифртекста – d операций получения остатка от деления для каждого из полиномов $(\pi(x) - mod\ p, \rho(x) - mod\ q)$. Следовательно, в криптосистеме Доминго-Феррера для шифрования количество операций получения остатка от деления $ModCount$ определяется по формуле (7).

$$ModCount = 2d + 1, \tag{7}$$

где d – степень полинома представления шифртекста.

В результате проведенной оценки сложности видно, что операция шифрования в криптосистеме Доминго-Феррера состоит из:

- $d-1$ операций выбора случайного элемента;
- $d-2$ операций сложения;
- 1 операция вычитания;
- $2\cdot\sum_{i=1}^{d} i$ умножений;
- $2d+1$ операций получения остатка от деления.

Приведенные результаты оценки количества математических операций при шифровании подтверждены экспериментальными исследованиями разработанной в рамках [11] реализации шифра Доминго-Феррера на языке C#.

Согласно [12] для степени полинома представления шифртекста $d = 2048$ суммарное число процессорных тактов при шифровании составит 17037369. На одном ядре процессора с тактовой частотой 2 ГГц оценочное время шифрования составит 8,5 мс.

Количество выполняемых операций умножения при расшифровании

На начальном этапе расшифрования в криптосистеме Доминго-Феррера вычисляются значения $A_p(x)$ и $A_q(x)$ по формулам (8) и (9) соответственно [13, 14].

$$A_p(x) = \left(b_d \cdot (r_p^{-1})^d\, x^d + \cdots + b_1 \cdot (r_p^{-1})\, x\right) mod\ p, \qquad (8)$$

где b – коэффициенты полинома $\pi(x)$ шифртекста.

$$A_q(x) = \left(b_d \cdot (r_q^{-1})^d\, x^d + \cdots + b_1 \cdot (r_q^{-1})\, x\right) mod\ q, \qquad (9)$$

где b – коэффициенты полинома $\rho(x)$ шифртекста.

Поскольку ключ не изменяется, подразумевается, что мультипликативные обратные ключа (r_p^{-1}, r_q^{-1}) были вычислены на этапе его генерации и сохранены в памяти, так что при расшифровании их вычисление не требуется.

Кроме того, в процессе расшифрования криптосистемой Доминго-Феррера операция умножения используется в Китайской теореме об остатках (5 раз) и в расширенном алгоритме Евклида (в среднем 42 раза для значений p и q до 10000).

Таким образом, при расшифровании в криптосистеме Доминго-Феррера количество умножений $MulCount$ определяется по формуле (10).

$$MulCount = 2 \cdot \sum_{i=1}^{d} i + 5 + 42, \quad (10)$$

где d – степень полинома представления шифртекста.

График зависимости количества операций умножения от степени полинома представления шифртекста приведен на рис. 3.

Рис. 3. Зависимость количества операций умножения от степени полинома представления шифртекста при расшифровании

Количество выполняемых операций деления при расшифровании

Деление при расшифровании криптосистемой Доминго-Феррера используется в Китайской теореме об остатках [15–17]. Количество делений не зависит от степени полинома представления шифртекста d, однако зависит от выбранных параметров p и q при вычислении мультипликативных обратных.

Количество выполняемых операций сложения и вычитания при расшифровании

В криптосистеме Доминго-Феррера для $A_p(x)$ вычисляется сумма всех коэффициентов M_p по формуле (11).

$$M_p = \sum_{i=1}^{d} b_i \ mod\ p, \quad (11)$$

где b – коэффициенты полинома $A_p(x)$.

Аналогично сумме M_p вычисляется сумма M_q по формуле (12).

$$M_q = \sum_{i=1}^{d} b_i \, mod \, q, \tag{12}$$

где b – коэффициенты полинома $A_q(x)$.

Из формул (11) и (12) следует, что для вычисления сумм M_p и M_q суммируются все коэффициенты полиномов $A_p(x)$ и $A_q(x)$ соответственно. Для сложения d слагаемых требуется выполнить $d-1$ операций сложения. Таким образом, в криптосистеме Доминго-Феррера в процессе вычисления сумм M_p и M_q выполняется $2(d-1)$ операций сложения.

Кроме того, в процессе расшифрования криптосистемой Доминго-Феррера операция сложения используется в Китайской теореме об остатках (2 раза), а в расширенном алгоритме Евклида используется операция вычитания (в среднем 42 раза для значений p и q до 10000). Таким образом, в криптосистеме Доминго-Феррера в процессе расшифрования выполняется $2(d-1)+2+42$ операций сложения и вычитания.

Количество выполняемых операций получения остатка от деления при расшифровании

Из формул (10) и (11) следует, что для вычисления значений $A_p(x)$ и $A_q(x)$ и затем, для подсчёта сумм M_p и M_q (формулы (11) и (12)) в криптосистеме выполняются операции получения остатка от деления на p и q соответственно. Кроме того, при вычислении открытого текста по Китайской теореме об остатках также выполняется одна операция получения остатка от деления на n. Таким образом, в криптосистеме Доминго-Феррера выполняется $2 \cdot d + 3$ операций получения остатка от деления.

В результате проведенной оценки сложности видно, что операция расшифрования в криптосистеме Доминго-Феррера состоит из:

- $2(d-1)+2$ операций сложения;
- 42 операций вычитания;
- $2 \cdot \sum_{i=1}^{d} i + 5 + 42$ умножений;
- $2d + 3$ операций получения остатка от деления.

Приведенные результаты оценки количества математических операций при расшифровании подтверждены экспериментальными исследованиями разработанной в работе [11] реализации шифра Доминго-Феррера на языке C#.

Согласно [12] для степени полинома представления шифртекста $d = 2048$ суммарное число процессорных тактов при расшифровании составит 17036373. На одном ядре процессора с тактовой частотой 2 ГГц оценочное время шифрования одного блока длиной 64 бита составит 8,5 мс.

Выводы

В рамках исследования оценена сложность выполнения операций шифрования и расшифрования гомоморфной криптосистемой Доминго-Феррера, приведены формулы расчёта количества операций, а также графики, отражающие зависимости количества операций от выбранных параметров шифра. Полученные значения сложности шифрования и расшифрования подтверждены экспериментальными исследованиями. Оценочное время шифрования и расшифрования 1 блока размером 64 бита криптосистемой Доминго-Феррера со степенью полинома представления шифртекста $d = 2048$ и значением модуля n до 10^8 на одном ядре процессора с тактовой частотой 2 ГГц оказалось примерно равным (разница $< 2 \cdot 10^{-7}$ мс) и составило 8,5 мс.

Исходя из проведенных исследований, можно прийти к заключению, что в криптосистеме Доминго-Феррера наиболее трудоёмкими являются заключительный этап шифрования (формулы (4) и (5)) и начальный этап расшифрования (формулы (8) и (9)), поскольку требуется возведение в степени от 1 до степени полинома представления шифртекста d значений ключа, что приводит к сложности операций шифрования и расшифрования $O(d^2)$, и в рамках реализации на языках программирования требует оптимизации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Domingo-Ferrer J.* A provably secure additive and multiplicative privacy homomorphism // International Conference on Information Security. – Berlin, Heidelberg : Springer Berlin Heidelberg, 2002. – P. 471-483.

2. *Hariss K., Noura H., Samhat A. E.* An efficient fully homomorphic symmetric encryption algorithm // Multimedia Tools and Applications. – 2020. – Vol. 79, No. 17. – p. 12139-12164.
3. *Wang H., Wang Z., Domingo-Ferrer J.* Anonymous and secure aggregation scheme in fog-based public cloud computing // Future Generation Computer Systems. – 2018. – Vol. 78. – P. 712-719.
4. *Панасенко С.П.* Алгоритмы шифрования. Специальный справочник. СПб.: БХВ-Петербург, 2009.
5. *Fan J., Vercauteren F.* Somewhat practical fully homomorphic encryption // Cryptology ePrint Archive. – 2012.
6. *Gentry C., Sahai A., Waters B.* Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based // Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. – Springer Berlin Heidelberg, 2013. – P. 75-92.
7. *Brakerski Z.* Fully homomorphic encryption without modulus switching from classical GapSVP //Annual cryptology conference. – Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. – P. 868-886.
8. *Brakerski Z., Gentry C., Vaikuntanathan V.* (Leveled) fully homomorphic encryption without bootstrapping // ACM Transactions on Computation Theory (TOCT). – 2014. – Vol. 6, No. 3. – P. 1-36.
9. *Смарт Н.* Алгоритмы возведения в степень // Криптография. – М.: Техносфера. – 2005.
10. *Данилкова Е. Р.* Теоретико-числовые алгоритмы в криптографии // Школа молодых ученых. – 2021. – С. 39-43.
11. *Бабенко Л.К., Стародубцев В.С.* Особенности реализации системы криптоанализа гомоморфных шифров, основанных на задаче факторизации чисел // Известия ЮФУ. Технические науки. – 2024. – № 2.
12. *Agner F.* Optimizing software in C++: An optimization guide for Windows, Linux and Mac platforms. – 2020.
13. *Трепачева А.В.* Улучшенная атака по известным открытым текстам на гомоморфную криптосистему Доминго-Феррера // Труды Института системного программирования РАН. – 2014. – Т. 26, № 5. – С. 83-98.
14. *Alabdulatif A., Kaosar M.* Privacy preserving cloud computation using Domingo-Ferrer scheme // Journal of King Saud University-Computer and Information Sciences. – 2016. – Vol. 28, No. 1. – P. 27-36.
15. *Ишмухаметов Ш.Т.* Методы факторизации натуральных чисел: учебное пособие. – 2011.
16. *Нестеренко А.* Введение в современную криптографию. Теоретико-числовые алгоритмы: курс лекций. – 2011.
17. *Осипов Н.Н. и др.* Теория чисел. – Красноярск: СФУ Институт космических и информационных технологий, 2008.

УДК 004.056.5

А.С. Белов[1], М.М. Добрышин[1] , А.Ф. Супрун[2]

[1]Россия, г. Орёл, Академия ФСО России

[2]Россия, г. Санкт-Петербург, СПБПУ

ИНТЕГРАЦИЯ И ДОПОЛНЕНИЕ ТЕРМИНОЛОГИИ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ С УЧЕТОМ СОВРЕМЕННЫХ ТЕНДЕНЦИЙ

Совершенствование информационных технологий и средств их реализующих способствуют разработки широкого перечня правовых актов, регламентирующих порядок функционирования и обеспечения информационной безопасности. Особенности функционирования инфотелекоммуникационных систем требуют формирования собственных подходов обеспечения их информационной безопасности, а также уточнения и дополнения специального терминологического базиса. С целью унификации терминологии на основании актуальных нормативных документов и практического опыта обеспечения информационной безопасности сформулирована, адаптирована и дополнена система терминов и определений.

Ключевые слова: *информационная безопасность; терминология; корпоративная сеть связи.*

The improvement of information technologies and the means of implementing them contribute to the development of a wide range of legal acts regulating the functioning and ensuring information security. The specifics of the functioning of infotelecommunication systems require the formation of their own approaches to ensuring their information security, as well as clarifying and supplementing a special terminology basis. In order to unify terminology on the basis of relevant regulatory documents and practical experience in ensuring information security, a system of terms and definitions has been formulated, adapted and supplemented.

Keywords: *information security; terminology; corporate communication network.*

Теория информационной безопасности (ИБ), как и все прикладные области знаний, появилась, формировалась и развивается на основе возникновения противоречий в практике, а именно из-за ущерба от действий нарушителей ИБ [1–5].

Для понимания сущности, содержании основных элементов и этапов развития теории ИБ, проведен анализ ряда работ по философии в области развития теории ИБ, который позволил разделить структуру теории на ядро теории, теоретический и эмпирический базисы [6, 7].

В качества *ядра теории* выступают: система терминов и определений (терминология), концептуальная модель (система взглядов) процесса обеспечения ИБ, объединяющая различные объекты защиты и связи между ними (законы изменений свойств элементов), а также ограничения и допущения.

Теоретический базис теории ИБ включает: совокупность правил обработки знаний и способов аргументации (методы, математическо-логический аппарат и др.) о событиях/инцидентах ИБ; научные проблемы; гипотезы и суждения предположительного характера с неопределенным значением истинности; наглядные модели протекающих процессов реализации компьютерных атак (КА) и функционирования объектов защиты в условиях КА, закономерности – т.е. обобщенные знания об изменении свойств изучаемых объектов; набор правил формирования моделей.

Эмпирический базис обобщает: совокупность операционных структур – набор правил и начальных условий измерений, моделирования, проведения экспериментов с реальным объектом; набор или протокольное описание фактов (событий/инцидентов ИБ); совокупность высказываний, описывающих результаты опытов, наблюдений, экспериментов, измерений параметров и интерпретаций результатов.

Совершенствование телекоммуникационного оборудования, информационных технологий, повышение требований абонентов к количеству и качеству предоставляемых услуг связи, достижения в других областях знаний, а также развитие средств и способов деструктивного воздействия имеющегося у нарушителей ИБ, порождают противоречия в практике, для устранения которых разрабатываются новые способы и методы получения, обработки данных и принятия решения. Расширение научного инструментария требует уточнения (дополнения) терминологии и адаптации концептуальных моделей к новым условиям.

Ретроспективный анализ уставных документов, регламентирующих ИБ, свидетельствует о поэтапном расширении исследуемой области и усложнение моделей идеализированного объекта, а именно пере-

ход от системы свойств защищаемой информации (целостность, доступность, конфиденциальность) к системе свойств характеризующих качество предоставляемой услуги связи в условиях воздействий на защищаемую информацию, средства обработки, хранения и передачи информации, сеть связи и применяемые информационные технологии.

Объединение нескольких направлений развития техники привел к наличию нескольких групп определений, не противоречащих друг другу, но сужающих область практической деятельности, например.

Информационная безопасность (information security continuity) – сохранение конфиденциальности, целостности и доступности информации (также могут рассматриваться и другие свойства: подлинность, подотчетность, неотказуемость и достоверность) [8].

Безопасность информационной технологии – состояние защищенности информационной технологии, при котором обеспечивается выполнение изделием, реализующим информационную технологию, предписанных функций без нарушений безопасности обрабатываемой информации [9].

Безопасность сети электросвязи – способность сети электросвязи противодействовать определенному множеству угроз, преднамеренных или непреднамеренных дестабилизирующих воздействий на входящие в состав сети средства, линии связи и технологические процессы, которые могут привести к ухудшению качества услуг, предоставляемых сетью электросвязи [10].

Событие (информационной) безопасности – зафиксированное состояние информационной (автоматизированной) системы, сетевого, телекоммуникационного, коммуникационного, иного прикладного сервиса или информационно-телекоммуникационной сети, указывающее на возможное нарушение безопасности информации, сбой средств защиты информации, или ситуацию, которая может быть значимой для безопасности информации [11].

Нарушение безопасности информации (в информационных (автоматизированных) системах) – совокупность событий безопасности и (или) иных данных мониторинга, указывающая на возможное нарушение конфиденциальности, целостности и доступности информации, нарушение принятой политики безопасности или наличие уязвимости [12].

Сравнение определений «*События информационной безопасности*» и «*Информационная безопасность*» показывает, что термин распространяется за пределы свойств защищаемой информации и охватывает термины «*Безопасность информационной технологии*» и «*Безопасность сети электросвязи».*

Подобные несоответствия существуют и для других терминов, что способно негативно влиять на адекватность формирования политики ИБ и, как следствие, возникновению ущерба от инцидентов ИБ.

Для устранения выявленного несоответствия на основе обобщения, адаптации и уточнения терминов, актуальных документов [8–16], тенденций развития средств и способов реализации КА, тенденций обеспечения ИБ, сформулирована терминология, позволяющая описать процесс обеспечения ИБ корпоративной сети связи с учетом системного представления протекающих процессов предоставления услуг связи.

Терминология обеспечения ИБ сети связи и информационных систем

Сформулированная терминология объединяет группы определений описывающих объекты защиты и протекающие процессы при обеспечении ИБ, объединенные единым замыслом (рис. 1).

Рис. 1. Система терминов, описывающих обеспечение ИБ корпоративной сети связи

Корпоративная сеть связи (КСС) – территориально распределенная информационная система, создаваемая на базе уже существующих сетей (локальных сетевых структур, сетей связи общего пользования, сети Интернет, сетей связи операторов связи) и узлов, предоставляющая абонентам сети заданный набор услуг связи и информационного обеспечения с требуемым качеством.

Актив – материальные и нематериальные ценности, используемые для достижения целей организации и являющиеся объектом защиты (к активам относятся услуги связи, информационная система, ресурсы, процессы) на всех этапах жизненного цикла.

Под *объектом защиты КСС* понимается совокупность аппаратных, программных, программно-аппаратных средств и комплексов, выполняющих функции обработки, хранения и передачи данных.

Риск ИБ КСС – представление возможного события и его последствий на достижение целей функционирования КСС.

Средство обработки информации – совокупность автономных устройств сбора, накопления, передачи, обработки и представления информации.

Информационная технология – процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способ осуществления таких процессов и методов.

Термины, описывающие процесс обеспечения ИБ КСС

Информационная безопасность КСС – соответствие значений параметров, характеризующих безопасность информации, информационных технологий, средств обработки, хранения и передачи информации требуемым значениям.

Уровень информационной безопасности КСС – обобщенный показатель (мера), характеризующий (характеризующая) соответствие значений параметров безопасности информации, информационных технологий, средств обработки, хранения и передачи информации, а также архитектурные решения, применяемые для предоставления пользователям услуг связи с заданным качеством.

Защищенность – мера, характеризующая долю уязвимостей объекта защиты, которыми может воспользоваться злоумышленник, для нанесения ущерба активам КСС, посредствам реализации известных КА, от общего количества уязвимостей объекта защиты КСС.

Уязвимость КСС – недостаток программного (программно-технического) средства или информационной системы в целом, которым (которая) может быть использована для реализации угроз ИБ.

Поверхность защиты – взаимоувязанная совокупность уязвимостей объекта защиты, средств и механизмов защиты и активов.

Политика информационной безопасности – комплекс мер, процедур, правил и принципов, направленных на недопущение или минимизацию ущерба активам КСС при реализации угроз ИБ.

Термины, описывающие дестабилизирующие воздействия на КСС с точки зрения ИБ

Нарушитель информационной безопасности – физическое лицо или логический объект, преднамеренные или непреднамеренные действия которого повлекли нарушение информационной безопасности.

Угроза ИБ – совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения ИБ.

Инцидент информационной безопасности – непредвиденное или нежелательное событие (группа событий) ИБ, которое привело (может привести) к нарушению функционирования объекта защиты или возникновению угроз ИБ.

Компьютерный инцидент – факт нарушения и (или) прекращения функционирования объекта защиты, используемого для организации взаимодействия информационных ресурсов, и (или) нарушения ИБ, в том числе произошедший в результате компьютерной атаки.

Компьютерная атака (КА) – целенаправленное воздействие программных и (или) программно-аппаратных средств на объект защиты в целях нарушения и (или) прекращения его функционирования и (или) создания угрозы ИБ.

Событие ИБ – зафиксированное состояние объекта защиты, применяемых информационных технологий, указывающее на возможное нарушение ИБ, или ситуацию, которая может быть значимой для защищаемых активов.

Ущерб – определенные или неопределенные последствия негативного влияния нарушения ИБ на достижение целей функционирования КСС.

Поверхность компьютерной атаки – количество способов, которыми злоумышленник может воспользоваться, чтобы нанести ущерб активам.

Траектория реализации компьютерной атаки – последовательность эксплуатации одной или группы уязвимостей объекта защиты, одной или группой КА, с целью нанесения ущерба активу КСС.

Тактика проведения компьютерной атаки – совокупность приемов и способов действий, используемых для проведения КА.

Техника проведения компьютерной атаки – совокупность и порядок действий, используемых для проведения КА в рамках соответствующих тактик.

Представленная терминология за счет адаптации и уточнения известных определений и дополнения новыми, позволяет описать процесс обеспечения ИБ с учетом текущего уровня развития техники и устранить неоднозначное трактование отдельных связей и протекающих процессов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Зегжда П.Д., Зегжда, А.Ф. Супрун, [и др.].* Методический подход к построению моделей прогнозирования показателей свойств систем информационной безопасности // Проблемы информационной безопасности. Компьютерные системы. – 2019. – № 4. – С. 45-49.
2. *Anisimov V.G., P.D. Zegzhda, [и др.].* Indices of the effectiveness of information protection in an information interaction system for controlling complex distributed organizational objects//Automatic Control and Computer Sciences. – 2017. – Vol. 51, No. 8. – P. 824-828.
3. *Сауренко Т.Н., Анисимов Е.Г. [и др.].* Прогнозирование инцидентов информационной безопасности/ Проблемы информационной безопасности. Компьютерные системы. – 2019. – № 3. – С. 24-28.
4. *Anisimov V.G., Zegzhda P.D., Suprun A.F., Anisimov E.G.* The problem of innovative development of information security systems in the transport sector // Automatic Control and Computer Sciences. – 2018. – Vol. 52, No. 8. – P. 1105-1110.
5. *Anisimov V.G., Anisimov E.G., Saurenko T.N., Zotova E.A.* Models of forecasting destructive influence risks for information processes in management systems // Information and Control Systems. – 2019. – No. 5 (102). – P. 18-23.
6. *Кузьнецов И.В.* Структура научной теории и структура объекта // Вопросы философии. – 1968. – № 5. – С. 1-12.

7. *Добрышин М.М.* Тенденции развития теории информационной безопасности в условиях динамического изменения парадигмы применения информационно-технических воздействий // Экономика и качество систем связи. – 2022. – № 1 (23). – С. 37-43.
8. ГОСТ Р ИСО/МЭК 27000-2021. Методы и средства обеспечения безопасности системы менеджмента информационной безопасности. Общий обзор и терминология.
9. ГОСТ Р 53114-2008. Защита информации. Обеспечение информационной безопасности в организации. Основные термины и определения
10. ГОСТ Р 53801-2010. Связь федеральная. Термины и определения.
11. ГОСТ Р 59547-2021. Защита информации. Мониторинг информационной безопасности. Общие положения.
12. ГОСТ Р 53729-2009. Качество услуги «Предоставление виртуальной частной сети (VPN)». Показатели качества.
13. ГОСТ 56546-2015. Защита информации. Уязвимости информационных систем. Классификация уязвимостей информационных систем.
14. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения.
15. ГОСТ Р 59548-2022. Защита информации. Регистрация событий безопасности. Требования к регистрируемой информации.
16. Контроль защищенности и мониторинг информационной безопасности. АНО ДПО «Учебный центр «Эшелон» uc-echelon.ru.

УДК 004

В.В. Вилков, В.Д. Михайлова

Россия, г. Таганрог, Южный федеральный университет

АТАКИ НА ОБЛАЧНЫЕ СЕРВИСЫ И МЕРЫ ПО ЗАЩИТЕ ДАННЫХ

В данной статье рассматриваются типичные атаки на облачные сервисы, способы получения данных, включая фишинг, ошибки администрирования, влияющие на безопасность данных, а также уязвимости Docker образов. Особое внимание уделено ключам безопасности и алгоритмам шифрования. Представлены рекомендации по повышению уровня информационной безопасности.

Ключевые слова: *облачные сервисы; фишинг; пароль; Docker контейнер; ключи безопасности; шифрование; информационная безопасность.*

This article examines common attacks on cloud services, methods of data acquisition including phishing, administrative errors impacting data security, and vulnerabilities in Docker images. Special attention is given to security keys and encryption algorithms. Recommendations for enhancing information security levels are presented.

Keywords: *cloud services; phishing; password; Docker container; security keys; encryption; information security.*

Введение

С развитием технологий и активным использованием интернета вопрос безопасной передачи и хранения данных становится все более актуальным. Облачные сервисы предлагают эффективные решения для хранения данных, однако они также подвержены различным атакам. В данной статье будут рассмотрены основные типы атак на облачные сервисы и меры по защите данных.

1. Обзор облачных хранилищ

MinIO

MinIO является высокопроизводительным масштабируемым объектным хранилищем для хранения больших объемов неструктурированных данных, таких как фотографии, видео, лог-файлы, архи-

вы и контейнерные образы. Он разработан с учетом потребностей облачных приложений и предоставляет API, совместимые с Amazon S3, что позволяет легко интегрировать его в существующие системы и приложения.

Преимущества MinIO:

- **Совместимость с Amazon S3:** MinIO полностью совместим с Amazon S3, что обеспечивает легкую миграцию и интеграцию с существующими приложениями и сервисами.
- **Масштабируемость:** MinIO может быть легко масштабирован горизонтально, добавляя новые узлы к кластеру без простоя.
- **Высокая производительность:** благодаря использованию современных технологий хранения и обработки данных MinIO обеспечивает высокую скорость чтения и записи данных.
- **Безопасность:** MinIO поддерживает различные механизмы аутентификации и шифрования, включая TLS для защиты передачи данных и политик доступа на уровне объектов для контроля доступа к данным [1].

Amazon S3

Amazon S3 (Simple Storage Service) – это веб-сервис от Amazon Web Services (AWS) для хранения и извлечения данных в любое время и из любого места в интернете. Он предлагает масштабируемое, надежное и безопасное облачное хранилище для данных любого объема.

Преимущества Amazon S3:

- **Высокая доступность и надежность:** Amazon S3 автоматически реплицирует данные в нескольких зонах доступности (Availability Zones) для обеспечения высокой доступности и защиты данных от потерь.
- **Различные классы хранения:** S3 предлагает различные классы хранения, такие как S3 Standard, S3 Intelligent-Tiering, S3 Glacier и S3 Glacier Deep Archive, для различных требований по доступности и стоимости хранения.
- **Безопасность и контроль доступа:** Данные в Amazon S3 могут быть защищены с помощью политики бакетов (bucket policies), политики доступа на уровне объектов (object policies) и списков контроля доступа (ACLs).

- **Журналирование и мониторинг:** Amazon S3 предоставляет функции журналирования доступа (server access logging) и мониторинга с помощью AWS CloudTrail, что позволяет отслеживать запросы к объектам и бакетам [2].

2. Сравнение MinIO и Amazon S3

MinIO и Amazon S3 являются мощными инструментами для облачного хранения данных, однако у них есть свои отличия. MinIO предлагает высокую производительность и гибкость, особенно для частных облачных решений, в то время как Amazon S3 предоставляет высокую доступность, надежность и широкий спектр классов хранения для масштабируемых облачных приложений, сравнение представлено на табл. 1.

Таблица 1

Сравнение MinIO и Amazon S3

Характеристика	MinIO	Amazon S3
Совместимость	Полная совместимость с Amazon S3	Прямой сервис от AWS
Масштабируемость	Горизонтальная, без простоя	Высокая, с различными классами
Производительность	Высокая	Высокая
Безопасность	TLS, политики доступа	Политики бакетов, ACLs, шифрование
Журналирование и мониторинг	Ограниченные возможности	AWS CloudTrail, CloudWatch
Стоимость	Зависит от инфраструктуры	Платные тарифы AWS

3. Типичные атаки на облачные сервисы

Фишинг

Фишинг является одной из наиболее распространенных атак, направленных на получение конфиденциальных данных пользователей. Атакующие создают поддельные веб-сайты, которые выглядят как легитимные, чтобы обманом заставить пользователей ввести свои учетные данные.

Фишинговые атаки часто проводятся через электронную почту, где злоумышленники отправляют сообщения, имитирующие официальные уведомления от известных сервисов. Они могут включать ссылки на поддельные веб-сайты, где пользователи вводят свои логины и пароли, тем самым передавая их злоумышленникам. Кроме того, фишинг может осуществляться через SMS-сообщения и социальные сети [4].

Фишинг является одной из наиболее распространенных атак, направленных на получение конфиденциальных данных пользователей. Согласно отчету Hornetsecurity за 2024 год, фишинг составляет 39.6% всех email-угроз, а 94% вредоносного ПО доставляется через электронную почту. В Великобритании фишинговые атаки составляют 83% всех зарегистрированных кибератак [3].

Инструменты для фишинга

Для проведения фишинговых атак злоумышленники используют различные инструменты:

- **PhishTool:** Инструмент для создания поддельных веб-страниц.
- **Gophish:** Платформа для проведения фишинговых кампаний.
- **SET (Social Engineering Toolkit):** Набор инструментов для социальных инженерных атак.

Эти инструменты позволяют злоумышленникам легко создавать и распространять поддельные веб-сайты и фальшивые письма, направленные на получение учетных данных пользователей.

Атаки на уязвимости Docker образов

Docker образы могут содержать уязвимости, которые могут быть использованы злоумышленниками для компрометации системы. В интерфейсе Docker Desktop можно просмотреть информацию о существующих уязвимостях. Например, уязвимости в используемых библио-

теках или конфигурационных файлах могут быть обнаружены и устранены путем регулярного обновления образов и использования инструментов сканирования безопасности, таких как Trivy или Clair [5].

Trivy

Trivy – это инструмент для сканирования контейнеров на наличие уязвимостей. Он проверяет контейнерные образы на наличие известных уязвимостей в операционных системах и библиотеке приложений. Trivy может сканировать как локальные образы, так и образы в удаленных реестрах контейнеров.

Особенности Trivy:

- Легкость использования: Быстрая установка и простота в использовании.
- Широкая база данных: Trivy использует базу данных уязвимостей от различных источников, включая Национальную базу данных уязвимостей (NVD) и базы данных уязвимостей поставщиков.
- Высокая скорость сканирования: Trivy быстро сканирует образы контейнеров и выдает результаты с детальной информацией о найденных уязвимостях.

Clair

Clair – это статический анализатор уязвимостей для контейнерных образов. Он интегрируется с различными реестрами контейнеров и автоматически сканирует новые образы на наличие уязвимостей.

Особенности Clair:

- Интеграция с реестрами: Clair легко интегрируется с реестрами контейнеров, такими как Docker Hub и Quay.io.
- Расширяемая архитектура: Clair поддерживает плагины для добавления новых источников данных об уязвимостях.
- API для автоматизации: Clair предоставляет RESTful API, что позволяет автоматизировать процесс сканирования уязвимостей.

Уязвимости могут возникать из-за использования устаревших компонентов или неправильной конфигурации Docker образов. Например, незащищенные порты или неправильно настроенные права доступа могут позволить злоумышленникам получить доступ к контейнерам. Для минимизации рисков рекомендуется использовать ми-

нимально необходимые права доступа, регулярно обновлять образы и проводить аудит безопасности. Согласно отчету Orca Security за 2024 год, 21% организаций имеют хотя бы одно общедоступное хранилище данных с чувствительной информацией, что увеличивает риск утечки данных и атак на цепочку поставок [6].

Ошибки администрирования

Ошибки в настройке облачных сервисов могут привести к утечке данных или несанкционированному доступу. Например, неправильные настройки прав доступа могут позволить пользователям не только читать файлы, но и удалять или изменять их. Для минимизации рисков рекомендуется проводить регулярные аудиты безопасности и использовать принципы наименьших привилегий.

Неправильное управление учетными записями и привилегиями может привести к серьезным проблемам безопасности. Например, если администратор не ограничивает доступ к критически важным ресурсам, злоумышленник может получить полный контроль над системой. Регулярное проведение аудитов и мониторинг действий пользователей помогут выявить и предотвратить подобные ошибки.

Атаки на слабые пароли и методы аутентификации

Использование слабых паролей и ненадежных методов аутентификации делает системы уязвимыми для атак. Злоумышленники могут использовать атаки методом подбора паролей (brute force) или атаки с использованием словарей (dictionary attacks) для получения доступа к учетным записям. Для защиты от таких атак рекомендуется использовать сложные пароли и многофакторную аутентификацию.

Согласно статистике, использование слабых паролей остается одной из основных причин утечек данных. Более 80% взломов аккаунтов происходит из-за использования слабых или украденных паролей [7]. Пароли, такие как "123456" или "password", остаются одними из самых распространенных и легко поддающихся взлому. Для защиты рекомендуется использовать сложные пароли, включающие комбинации букв, цифр и специальных символов, длиной не менее 12 символов. Кроме того, рекомендуется использовать менеджеры паролей и двухфакторную аутентификацию [8].

4. Ключи безопасности и шифрование

Открытые и закрытые ключи

Облачные сервисы, такие как MinIO, используют пару ключей (открытый и закрытый) для аутентификации и шифрования данных. Открытый ключ используется для шифрования данных, в то время как закрытый ключ используется для их дешифрования. MinIO поддерживает различные алгоритмы шифрования, включая AES (Advanced Encryption Standard) и RSA (Rivest-Shamir-Adleman). Эти алгоритмы обеспечивают высокий уровень безопасности при передаче и хранении данных.

Алгоритмы шифрования обеспечивают защиту данных, превращая их в зашифрованный текст, который может быть расшифрован только с использованием правильного ключа. AES является симметричным алгоритмом, что означает, что один и тот же ключ используется для шифрования и дешифрования данных. RSA, в свою очередь, является асимметричным алгоритмом, который использует пару ключей – открытый для шифрования и закрытый для дешифрования.

Применение ключей в практике

Применение шифрования на основе ключей позволяет защитить данные как при их передаче, так и при хранении. Например, данные, передаваемые между клиентом и сервером, могут быть зашифрованы с использованием TLS (Transport Layer Security), чтобы предотвратить их перехват злоумышленниками. TLS использует сертификаты для установления защищенного соединения и шифрования данных.

Кроме того, облачные сервисы могут использовать механизмы управления ключами, такие как AWS Key Management Service (KMS), для автоматизации управления ключами и повышения безопасности. KMS позволяет создавать, управлять и использовать ключи шифрования для защиты данных в облаке. Интеграция с KMS упрощает процесс шифрования данных и обеспечивает высокий уровень защиты.

5. Рекомендации по повышению уровня информационной безопасности

Обучение пользователей

Одной из ключевых мер по защите данных является регулярное обучение пользователей основам информационной безопасности. Пользователи должны быть осведомлены о возможных угрозах, таких как фишинг и социальная инженерия, и уметь распознавать подозрительные сообщения и ссылки. Обучение должно включать примеры реальных атак и рекомендации по безопасному поведению в сети.

Регулярное проведение тренингов и семинаров поможет пользователям оставаться в курсе последних угроз и методов защиты.

Регулярные обновления и патчи

Регулярные обновления программного обеспечения и установка патчей безопасности являются важными мерами по защите систем от уязвимостей. Злоумышленники часто используют известные уязвимости в программном обеспечении для проведения атак. Обновление систем и приложений до последних версий помогает закрыть уязвимости и защитить системы от атак.

Заключение

Безопасность облачных сервисов является критически важной для защиты конфиденциальных данных. Регулярные обновления и мониторинг уязвимостей, использование многофакторной аутентификации, а также правильные настройки прав доступа помогут защитить системы от атак. Внедрение современных алгоритмов шифрования и обучение пользователей распознаванию фишинговых атак также играют ключевую роль в обеспечении информационной безопасности.

Применение рекомендаций, изложенных в данной статье, поможет компаниям повысить уровень защиты своих облачных сервисов и обеспечить безопасность данных. Важно помнить, что информационная безопасность является непрерывным процессом, требующим постоянного внимания и улучшений.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Документация MinIO [Электронный ресурс] // min.io: [сайт]. – URL: https://min.io/docs/minio/linux/index.html (дата обращения: 05.07.2024).
2. Документация Amazon S3 [Электронный ресурс] // docs.aws.amazon.com: [сайт]. – URL: https://docs.aws.amazon.com/s3/ (дата обращения: 12.07.2024).
3. Статистика кибератак 2024 года // cobalt.io: [сайт]. – URL: https://www.cobalt.io/ blog/cybersecurity-statistics-2024 (дата обращения: 10.07.2024).
4. Proofpoint State of the Phish Report 2024 [Электронный ресурс] // norton.com: [сайт]. – URL: https://norton.com/blog/emerging-threats/2024-predictions (дата обращения: 18.07.2024).
5. Документация Docker [Электронный ресурс] // docs.docker.com: [сайт]. – URL: https://docs.docker.com/ (дата обращения: 25.05.2024).
6. Orca Security [Электронный ресурс] // orca.security: [сайт]. – URL: https://orca.security/lp/2024-state-of-cloud-security-report/#the-report (дата обращения: 18.07.2024).
7. Expert Insights Identity and Access Security Stats 2024 [Электронный ресурс] // expertinsights.com: [сайт]. – URL: https://expertinsights.com/insights/ 50-identity-and-access-security-stats-you-should-know/ (дата обращения: 18.07.2024).
8. Norton Password Statistics 2024 [Электронный ресурс] // norton.com: [сайт]. – URL: https://norton.com/blog/privacy/password-statistics (дата обращения: 18.07.2024).

УДК 004.415.52

А.Э. Волков, Е.В. Карачанская

Россия, г. Хабаровск, Дальневосточный государственный университет путей сообщений

КОНСТРУКТИВНЫЕ DOM-BASED XSS УЯЗВИМОСТИ КОДОВОЙ БАЗЫ SPA. АНАЛИЗ И РЕКОМЕНДАЦИИ К ИХ УСТРАНЕНИЮ

Проведено исследование сущности DOM-based XSS атак и связанных с ними уязвимостей кодовой базы SPA. Рассмотрены в общем плане суть и виды XSS атак. Описаны основные особенности и места исполнения DOM-based XSS атак, частично рассмотрены наиболее распространенные примеры подобных уязвимостей, и основные причины возникновения подобного рода уязвимостей.

Ключевые слова: *SPA; XSS; DOM-based XSS; Анализ уязвимостей; Кибербезопасность*

A study was conducted of the essence of DOM-based XSS attacks and the associated vulnerabilities of the SPA code base. The essence and types of XSS attacks are considered in general terms. The main features and places of execution of DOM-based XSS attacks were described, and the most common examples of such vulnerabilities were partially considered. The main reasons for the occurrence of this kind of vulnerabilities are described.

Keywords: *SPA; XSS; DOM-based XSS; Vulnerability analysis; Cybersecurity*

Разработчики веб-приложений с каждым днем все больше и больше сталкиваются с необходимостью обеспечения безопасности разрабатываемых проектов из-за растущего количества угроз со стороны злоумышленников, которые изобретают новые или улучшают старые способы проникновения в систему, хищения данных пользователей и так далее. Параллельно с этим не перестают появляться более совершенные продукты, подходы, принципы для защиты веб-приложений и анализа их уязвимостей. Несмотря на то, что уже многим известны эти продукты или решения, столько книг и материалов, описывающих те или иные уязвимости (угрозы) и способы их устра-

нения (защиты), но, к сожалению, мы нередко видим в новостных порталах сообщения об очередной утечке данных, проникновения злоумышленников в систему, искажения или удаления сведений из баз данных, порой, крупных и мировых компаний.

Если в крупных компаниях создаются целые отделы под обеспечение безопасности своих приложений и там уровень защищенности и развитости подходов к безопасной разработке наиболее высок, то в компаниях поменьше или менее ответственных в этом плане, груз защиты приложения ложится исключительно на разработчиков, которые не осведомлены об аспектах обеспечения безопасности приложений в той же мере, как и профильные специалисты данной области.

Как правило, обнаружение в приложении уязвимости для проведения той или иной атаки, является следствием неосведомленности разработчика о способах ее осуществления и, соответственно, не устранением данной уязвимости.

Целью работы является рассмотрение конструктивных особенностей разработки механизмов обработки ввода и вывода пользовательских данных, как наиболее уязвимых мест для проведения атаки XSS в DOM-модели (далее DOM-based XSS). Для этого сначала установим сущность XSS-атак, а затем проведем анализ уязвимых мест и конструкций в кодовой базе приложения и способов их устранения.

Межсайтовый скриптинг (XSS) – это способ атаки, включающий в себя передачу кода, предоставленного злоумышленником, в экземпляр браузера пользователя. В свою очередь, браузер может быть стандартным клиентом веб-браузера или объектом браузера, встроенным в программный продукт. Сам код в подавляющем большинстве случаев пишется на HTML/JavaScript.

Злоумышленник катализирует исполнение вредоносного кода (например, путем введения его в URL строку браузера), код в свою очередь исполняется в контексте безопасности (или зоне) хостингового веб-сайта. Благодаря этому уровню привилегий код потенциально может читать, изменять и передавать любые конфиденциальные данные, доступные браузеру. У пользователя в ходе проведения XSS-атаки может быть взломана учетная запись, его браузер перенаправлен на сайт злоумышленника или, возможно, показан мошеннический контент, доставленный веб-сайтом, который он посещает.

Как правило среди XSS-атак выделяют следующие:

1. Stored XSS. В данном случае вредоносный код попадает в клиентское приложения от сервера, а именно из базы данных.
2. Reflected XSS. Клиентское приложение, в ходе запроса к серверу для получения статичной HTML-страницы, после введения злоумышленником данных в URL запроса принимает «зараженную» страницу и исполняет вредоносный код на своей стороне.
3. DOM-based XSS. Уязвимость находится в клиентской части приложения, то есть, в данном случае, нет необходимости отправлять запрос на сервер, весь код исполняется на стороне клиента.

Отражение первых двух видов атак, как правило, в компетенции серверной части приложения, на него ложатся обязательства по правильной обработке данных от клиента, организации безопасного процесса формирования статических данных на основе полученных данных и так далее. При этом не стоит снимать ответственности в этом вопросе с клиентского приложения, ведь тут тоже можно организовать предварительную фильтрацию и экранирование (кодирование) потенциально опасных символов. Но, в современной разработке, принято возлагать на клиентскую часть веб-приложения, как можно меньше дополнительной логики, если этого можно избежать, в угоду оптимизации работы на устройствах конечных пользователей.

Однако существует тип XSS-атаки – DOM-based XSS, которая проводится в зоне работы, и как следствие, ответственности клиентской части приложения. Остановимся более детально на сущности данного вида XSS.

Обращаем внимание на наличие в наименовании рассматриваемого вида аббревиатуры «DOM».

DOM – это объектная модель документа, которую браузер создает в памяти компьютера на основании HTML-кода, полученного им от сервера. Иными словами, это представление HTML-документа в виде дерева тегов. Такое дерево необходимо для правильного отображения сайта и внесения изменений на страницах с помощью языка JavaScript.

XSS-уязвимости на основе DOM, как правило, возникают в момент получения данных JavaScript-ом из источника, контролируемого злоумышленником (например, URL-адрес, поля ввода), и пере-

дачей их в приемник, который поддерживает динамическое выполнение кода, например eval() или element.innerHTML. Данный аспект позволяет злоумышленникам выполнять вредоносный код JavaScript.

Для осуществления атаки такого рода, необходимо передать данные в источник, чтобы они распространялись на приемник и вызывали выполнение произвольного JavaScript.

Наиболее распространенным источником DOM-based XSS является URL-адрес, доступ к которому обычно осуществляется с помощью объекта window.location. Злоумышленник может создать ссылку для отправки жертвы на уязвимую страницу с полезной нагрузкой в строке запроса и фрагментами URL-адреса.

Для выявления опасных мест необходимо заострить внимание на две важные сущности данного процесса:

1. Источник (далее source) – место, через которое злоумышленник может передать вредоносный код, как правило, URL-адрес страницы (динамические части, Search Params и так далее).
2. Приемник (далее sink) – место, где в конечном итоге исполняется вредоносный код или конструкция языка, передача зараженного кода в которую, также вызовет его исполнение.

Определимся с вероятным перечнем source, мест откуда мы должны ожидать проникновение в приложение вредоносного кода злоумышленника:

1) document.URL;
2) document.URLUnencoded;
3) document.location, включая иные свойства этого объекта;
4) document.referrer;
5) window.location, включая иные свойства этого объекта.

Рассмотрим наиболее вероятные конструкции, которые выступают в качестве sink:

1. Прямая запись в HTML код:
   a. document.write(...);
   b. document.writeln(...);
   c. document.body.innerHtml...;
2. Прямое редактирование DOM:
   a. document.forms[0].action=...;
   b. document.attachEvent(...);

c. document.create(...);
d. document.execCommand(...);
e. document.body...(получение доступа к DOM-элементу через свойство document);
f. window.attachEvent(...);

3. Программное изменение URL-адреса:
   a. document.location=...;
   b. document.location. hostname=...;
   c. document.location.replace(…);
   d. document.location.assign(...);
   e. document.URL=...;
   f. window. navigate (...);
4. Открытие или изменение окна браузера:
   a. document.open(...);
   b. window.open(...);
   c. window.location.href=...;
5. Программная инициализация исполнения скрипта:
   a. eval(...);
   b. window.execScript(...);
   c. window.setInterval(...);
   d. window.setTimeout(...).

Разберем функционал, который встречается практически в любом веб-приложении. Одна из частых необходимостей – это заполнить DOM-элементы содержимым пользовательского ввода. Для этого не рекомендуется использовать не один из рассмотренных ранее вариантов конструкций языка, предназначенных для записи данных в DOM-элементы – они уязвимы для проведения DOM-based XSS.

На рис. 1 представлено сравнение способов выводов информации в DOM-элемент. В данном случае рассмотрен сценарий, когда из URL-адреса берется динамическая информация для вывода на экран, например, имя пользователя.

В рассмотренном сценарии важной задачей является обеспечить безопасность при обработке пользовательского ввода и вывода его в элемент. Те способы, которые промаркированы как «Не безопасные», таковыми являются вследствие того, что они, при получении данных, интерпретируют их как HTML, из-за чего, если вместо обычного имени там будет находится строка с условным содержи-

мым: «<script>alert(«XSS attack»)</script>», то в таком случае выведется на странице в браузере уведомление, с содержимым из аргумента функции «alert».

```
<script>
    let indexOfNameSearchParam = document.URL.indexOf("name=")+5;
    let userName = document.URL.substring(indexOfNameSearchParam,document.URL.length);

    // Не безопасный вывод
    document.write(userName);
    document.writeln(userName);
    document.body.innerHtml = userName;

    // Допустимый вывод
    document.body.innerText = userName;
    document.body.textContent = userName;
</script>
```

Рис. 1. Сравнение способов вывода информации в DOM-элемент

Способы, отмеченные как «Допустимы», при получении значения для добавления в DOM-элемент, вставляют его без интерпретации как HTML кода, следовательно, потенциально зараженное значение, которое содержит вредоносный код, просто выведется в DOM-элемент, без его исполнения.

Однако это вовсе не означает, что варианты под маркой «Не безопасные» запрещено использовать: при необходимости можно просто отфильтровать небезопасные символы или закодировать содержимое, используя стандартные инструменты JS.

Как можно заметить, точек входа для вредоносного кода злоумышленника не так уж и много, но мест где он потенциально можно исполниться существенно больше. Следовательно, разработчикам необходимо контролировать входные данные, их поток в приложении, способ их отображения на странице.

Поскольку применение перечисленных действий разработчиком приложений зависит от его квалификации в области информационной безопасности, поэтому существует необходимость разработки и внедрения методики, которая соберет в себя структурированную информацию по распространенным уязвимостям, поможет разработчикам в написании безопасного кода. Данная методика предполагает

создание правил для статического анализатора, который будет в процессе разработки подсвечивать потенциально опасные конструкции и операции в коде, что также заставит разработчиков обратить внимание на уязвимости в приложениях.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Матяш Е.Д., Никонов В.В., Иванова И.А.* Обеспечение безопасности Web-сайтов // Евразийский Союз Ученых. – 2016. – № 3-3 (24). – URL: https://cyberleninka.ru/article/n/obespechenie-bezopasnosti-web-saytov.
2. PortSwigger. DOM-based XSS [Электронный ресурс]. – URL: https://portswigger.net/web-security/cross-site-scripting/dom-based (дата обращения: 15.07.2024).
3. *Амелин Р.В.* Информационная безопасность. Учебно-методическое пособие по вопросам информационной безопасности [Электронный ресурс]. – URL: http://www.telecomlaw.ru/studyguides/amel_infobez.pdf. (дата обращения: 10.07.2024).
4. Open Web Application Security Project (OWASP). Top-10:2021 [Электронный ресурс]. – URL: https://owasp.org/Top10/A00_2021_Introduction (дата обращения: 06.07.2024).
5. *Крис Митчелл. Артем Конев*. Обеспечение безопасности веб-сайтов. // Australia: SophosLabs [Электронный ресурс]. – URL: http://help.yandex.ru/webmaster/protecting-sites/contents.xml (дата обращения: 09.07.2024).
6. Анализ защищенности веб-приложений [Электронный ресурс]. – URL: https://ddos-guard.net/ru/blog/analiz-zashchishchennosti-veb-prilozhenii (дата обращения: 09.07.2024).

УДК 004.89

О.Т. Данилова

Россия, г. Омск, Омский государственный технический университет

МОДЕЛИРОВАНИЕ ПРОЦЕССА ВОЗДЕЙСТВИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА АВТОНОМНУЮ КОМПЬЮТЕРНУЮ СИСТЕМУ

В работе на основе анализа структуры различных типов вредоносного программного обеспечения (ВПО) представлена разработка имитатора, наделенного определенным функционалом. Поставлен эксперимент с целью моделирования вредоносных воздействий на компьютерную систему, а также проведен анализ существующих инструментов для исследования ВПО, на основе которого сделан выбор необходимого набора программных средств для проведения исследования свойств разработанного исполняемого файла. Представлена и апробирована методика статического анализа ВПО. На основе полученных данных, сформированы рекомендации к защите от атак с использованием вредоносного программного обеспечения.

Ключевые слова: *имитатор вредоносного программного обеспечения; обфускация; декомпиляция.*

The paper presents a scheme of a malicious software simulator for remote access. For the experiment, a network with devices connected to it was built. An experiment was set up to simulate malicious effects on an autonomous computer system, and an analysis of the infected computer system was carried out and indicators of compromise were identified. Based on the data obtained, recommendations for protection against attacks using malicious software were formed.

Keywords: *malware simulator; obfuscation; decompilation.*

Вредоносное программное обеспечение – это обобщающий термин, который описывает любую вредоносную программу или код, наносящий вред системам. Внедрение ВПО в компьютерную сетевую среду имеет разные последствия в зависимости от замысла самого вредоносного ПО и схемы сети [1].

Специалист, привлеченный к расследованию компьютерных инцидентов, связанных с воздействием вредоносного программного обеспечения (ВПО), должен уметь создавать безопасную и изолиро-

ванную лабораторную среду для анализа вредоносных программ; извлекать метаданные, связанные с вредоносным ПО; определять взаимодействие вредоносных программ с системой [2].

Информация, полученная в процессе проведения исследования, позволяет:

- понять каким образом функционирует вредоносное программное обеспечение;
- атрибутировать индикаторы компрометации;
- определять способы не только восстановления зараженной системы, но и минимизации нанесенного ущерба [3–5].

В настоящей работе приводятся описание экспериментальной симуляции сценария, по которому проходит заражение компьютерной системы, а также последовательность действий позволяющих провести анализ результатов атаки.

При разработке модельного представления в работе учтено, что большинство атак с использованием вредоносного программного обеспечения можно описать с помощью модели Cyber Kill Chain [6–7]. В данную модель входят следующие этапы:

1. Получение нарушителем информации об архитектуре и топологии автоматизированной системы.
2. Выбор или разработка вредоносного программного обеспечения.
3. Доставка ВПО.
4. Заражение информационной системы.
5. Развертывание ВПО (например, вторичное закрепление в системе, повышение привилегий и так далее).
6. Полный контроль и управление зараженной системы.
7. Нанесение ущерба, за счет совершения противоправных действий.

Моделирование заключается в построении компьютерной сети с виртуализацией информационного ресурса, которая подвергается воздействию вредоносного программного обеспечения [8].

Для проведения эксперимента был разработан макет сети, схема которой представлена на рис. 1.

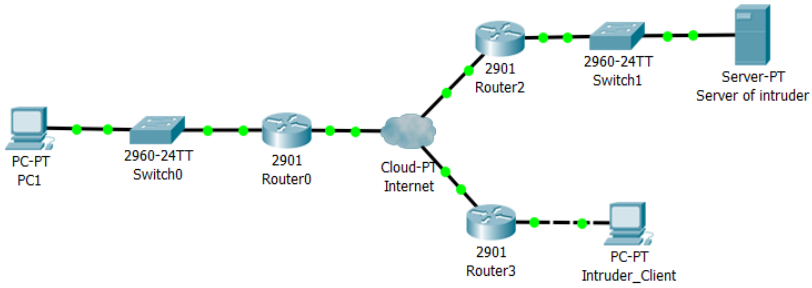


Рис. 1. Схема макета экспериментальной сети

В схеме представлены следующие ключевые устройства:

1. PC1 – персональный компьютер пользователя с операционной системой Windows 10. Имеется выход в сеть Интернет. Для контроля действий пользователей в сети настроены:

- аудит входа и выхода в систему;
- аудит доступа к общему сетевому ресурсу;
- аудит доступа к объектам;
- аудит отслеживания процессов.

Контроль сетевого трафика осуществляется встроенным брандмауэром операционной системы Windows, в котором активны правила для входящих и исходящих подключений. Устройство подвергается активации разработанной программной модели ВПО.

2. Server of intruder – сервер, который открыт для входящих подключений и способен передавать данные между подключенными клиентами. Исполняет роль посредника для передачи команд модели ВПО и данных нарушителю.

3. Intruder_Client – персональный компьютер нарушителя с операционной системой Windows 10. Имеется выход в интернет. На устройстве установлен программный клиент, поддерживающий сетевой протокол ММР. Программный клиент предназначен для отправки управляющих команд (через сервер «Server of intruder») модели ВПО, а также для получения запрашиваемых данных.

Легенда сценария инцидента следующая:

1) пользователь скачивает некоторый программный файл из сети Интернет, открывает его, ожидая обнаружить определенный функционал;

2) поскольку никакие окна не отображаются, сама программа выглядит нерабочей, то пользователь продолжает поиски нужного файла, а скачанное ВПО функционирует в фоновом режиме.

Для решения поставленных в работе задач был создан имитатор вредоносного программного обеспечения (ИВПО), поименованный как TRICKER.EXE. Функциональная блок-схема имитатора представлена на рис. 2.

Рис. 2. Блок-схема имитатора ВПО

Программная реализация ИВПО была исполнена в «Microsoft Visual Studio Community 2017» на языке C# с использованием WindowsFormsApp (библиотека моделирования окон пользовательского интерфейса). Исполняемый файл был впоследствии упакован с помощью программы PECompact, что позволило скрыть часть заголовков и сжать файл до меньшего размера.

После проведения эксперимента по внедрению вредоносного файла в систему было проведено в соответствии с рекомендациями стандарта ГОСТ Р 59712-2022 расследование и фиксация материалов, связанных с возникновением компьютерного инцидента [9].

Исследование имитатора ВПО включает следующие этапы:

1. Проверку образца с целью определения типа ВПО с помощью антивирусных сервисов.

2. Определение параметров исследуемого образца.

3. Исследование PE-заголовка.

4. Определение языка программирования, на котором написан исследуемый образец, и его разрядности.

5. Проверку на обфусцирование образца и в зависимости от результата проведение его деобфускации.

6. Подтверждение или опровержения факта наличия упаковки образца.

7. Проведение анализа функционала образца с использованием статического или динамического методов.

Оптимальным вариантом при проведении анализа ВПО является использование стендового компьютера и запущенной на нём виртуальной машиной. Организовав такое рабочее окружение, решается проблема с сетевым взаимодействием, так как все современные виртуальные машины способны эмулировать сеть между гостевой системой и хост системой, а в случае повреждения виртуальной машины стендовый компьютер примет на себя удар и поврежденным окажется именно он.

В соответствии с представленной схемой проверка ИВПО TRICKER.EXE в сервисе VirusTotal позволила атрибутировать его как троянскую программу типа BehavesLike.Win32.Generic.Im, которая способна проникнуть на любой компьютер с ОС Windows и оставаться незамеченной в течение длительного времени.

Для проверки файла на обфускацию и упаковку была выбрана утилита PEiD v0.95. Результаты анализа позволили определить факт упаковки средством PECompact 2.x – программы для сжатия исполняемых файлов для Windows с закрытым исходным кодом.

Распаковка исходного исполняемого файла была проведена с помощью утилиты UnPECompact2 v0.2. После распаковки файл TRICKER.EXE снова был проверен на вредоносность, но было отмечено, что результаты обнаружения отмечает меньшее количество антивирусных детекторов. После проверки исполняемый файл снова был проверен на наличие упаковки и обфускации, но на этом этапе утилита PEiD не обнаружила признаков упаковщика или обфускатора и позволила идентифицировать язык программирования компилятора – C# платформы .NET. Данный язык является интерпретируемым, а значит исполняемый файл может содержать всю нужную информацию для его декомпиляции. В качестве декомпилятора было выбрано ПО dotPeek от компании JetBrains.

Результаты, полученные в ходе проведения расследования вышеописанного инцидента позволили определить, что исполняемый файл, поименованный как TRICKER.EXE, исполняет в зараженной системе следующие действия:

1. Создает переменную «reg», которой устанавливает путь реестра к разделу автозагрузки.
2. Подключает элементы Timer и NotifyIcon.
3. По каждому отчету таймера, заменяет текст буфера обмена на «You've been compromised – Вы были скомпрометированы» (рис. 3).

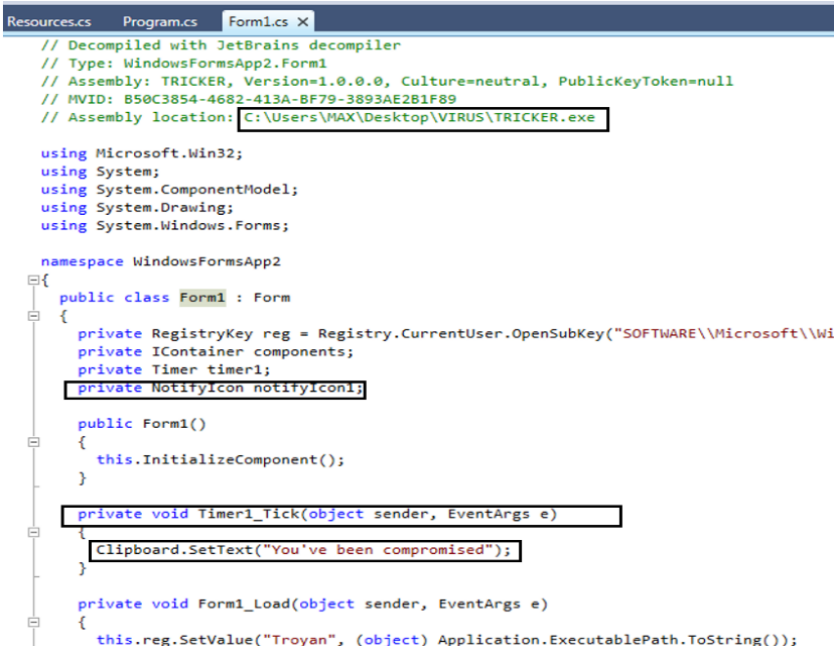


Рис. 3. Иллюстрация результатов замены текста буфера обмена

4. Задает переменной «reg» имя Troyan.
5. Включает таймер «timer1» для периодической проверки содержимого буфера обмена.
6. Скрывает значок приложения «notifyIcon1», а также все приложения с панели задач.

7. Включает таймер «timer1» для периодической проверки содержимого буфера обмена.

8. Скрывает значок приложения «notifyIcon1», а также все приложения с панели задач.

Иллюстрация полученных результатов представлена на рис. 4.

Рис. 4. Иллюстрация результатов анализа функционала имитатора ВПО

В качестве рекомендаций по обнаружению атак, подобных описанной в настоящей работе, можно предложить следующее:

- обработка событий безопасности должна осуществляться с применением SIEM систем, поскольку правильно настроенная SIEM система позволяет специалистам по информационной безопасности грамотно и своевременно реагировать на происходящие события;
- для проверки всех поступающих в систему файлов следует использовать среду безопасного тестирования SandBox;

- для выявления не детектируемого ВПО необходимо проводить анализ всех принятых файлов, чтобы снизить вероятность заражения информационной системы в случае выявления подозрительных событий;
- могут быть внедрены приманки для нарушителя (Deception), что позволит обнаружить и разобраться в методе и средствах атаки без ущерба для организации и прервать ее до наступления негативных последствий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Rains T.* Cybersecurity Threats, Malware Trends, and Strategies. – Birmingham: Packt Publishing, 2020. – 410 p.
2. *Самойлов В.Д.* Информационная безопасность в системе высшего образования России (компетентностный подход в подготовке специалистов). – М.: Компания КноРус, 2018. – 162 с.
3. *Sihwail R., Omar K., Ariffin K.A.Z.* A Survey on Malware Analysis Techniques: Static, Dynamic, Hybridand Memory Analysis // International Journal on Advanced Science, Engineering and Information Technology. – Vol. 8 (No. 4-2). – 2018. – P. 1662-1671.
4. *Sikorski M., Honig A.* Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software. – San Francisco: No Starch Press, 2012. – 800 p.
5. *Монапна К.А.* Анализ вредоносных программ. – М.: ДМК Пресс, 2019. – 452 с.
6. *Yadav M., Rao A.M.* Technical Aspects of Cyber Kill Chain // Security in Computing and Communications: Third International Symposium (SSCC 2015). – Kochi, India, 2015. – P. 438-452.
7. *Johnson T.A*. Cyber-security Protecting Critical Infrastructures from Cyber Attack and Cyber Warfare. – Boca Raton: CRC Press/Taylor & Francis Group, 2015. – 326 p.
8. *Белгородцев А.А., Данилова О.Т., Парыгин Е.Н.* Макет стенда-тренажера для имитации атак на информационные системы //Динамика систем, механизмов и машин: Материалы XIV Международной IEEE научно-технической конференции (Омск, 10–12 ноября). – 2020. – Том 8, № 2. – Омск: ОмГТУ 2020. – С. 93-97.
9. ГОСТ Р 59712-2022. Защита информации. Управление компьютерными инцидентами. Руководство по реагированию на компьютерные инциденты. – Введ. 2023–02–01. – М.: Российский институт стандартизации, 2022. – 20 с.

УДК 004.056.5

П.П. Дешевов, С.С. Укустов

Россия, г. Москва, МИРЭА

МОДЕЛЬ ЦИФРОВЫХ УДОСТОВЕРЕНИЙ ДЛЯ ВЕРИФИКАЦИИ НА ОСНОВЕ СИСТЕМ ДОКАЗАТЕЛЬСТВ С НУЛЕВЫМ РАЗГЛАШЕНИЕМ

В работе проводиться анализ существующих подходов аутентификации и авторизации пользователя на основе цифровых удостоверений. Анализируются неинтерактивные системы доказательств с нулевым разглашением, и как эти системы могут быть использованы для аутентификации и авторизации пользователя. На основе анализа данная работа предлагает новую модель данных цифровых удостоверений, которая не зависит от определенной платформы и определенной системы доказательств с нулевым разглашением.

Ключевые слова: *удостоверения с нулевым разглашением; аутентификация; авторизация; само-суверенная идентичность; доказательства с нулевым разглашением.*

This paper analyzes existing approaches to user authentication and authorization based on digital credentials. It analyzes non-interactive zero-knowledge proof systems, and how these systems can be utilized for user authentication and authorization. Based on this analysis, the paper proposes a new digital credential data model that is independent of any specific platform or particular zero-knowledge proof system.

Keywords: *zero-knowledge credentials; authentication; authorization; self-sovereign identity; zero-knowledge proofs.*

1. Актуальность исследования

За 2022 и 2023 было украдено больше записей в следствии инцидентов утечек данных чем за период с 2021 по 2011 года, больше 80% украденных записей составляют персональные данные пользователей информационных систем [1, 2]. Данная ситуация дает четкое понимание того, что современные подходы по обеспечению безопасности персональных данных пользователей далеки от идеала [3, 4]. Текущие инженерные решения предполагают накопление большого

количество неанонимизированных данных для обеспечения контроля доступа или высокого качества пользовательского опыта, что способствует утечкам. Минимизация сохраняемых данных о пользователе в едином цифровом сервисе представляется одним из наиболее перспективных подходов для уменьшения ущерба. Это уменьшит привлекательность злоумышленников осуществлять атаки на цифровой сервис.

В противовес текущей тенденции краж персональных данных, всё большую популярность набирают подходы, основанные на принципах само-суверенной идентичности (SSI) [5], который позволит пользователям, полностью контролировать свое цифровое представление о себе в цифровом пространстве.

2. Проблема

Существующие спецификации и стандарты не дают формального описания взаимодействия между основными компонентами архитектуры ИС для достижения всех принципов SSI. В тоже время текущие реализации ИС для аутентификации и авторизации пользователя не соответствуют всем принципам SSI.

В качестве первого шага для решения описанных выше проблем, данная работа предлагает описание формата цифровых удостоверений, которые позволят пользователю подтверждать утверждение о себе не раскрывая атрибутов удостоверения проверяющей стороне. В дополнение к этому описанная модель удостоверения не зависит от определенных платформ и конкретных неинтерактивных систем доказательств с нулевым разглашением.

3. Существующие методы и подходы

Подходы аутентификации и авторизации пользователя на основе удостоверения основываются на архитектуре треугольника доверия, в который входят следующие компоненты: эмитент – сущность, которая выдает удостоверения владельцу удостоверений; владелец удостоверений – сущность, которая хочет подтвердить с помощью удостоверения утверждение о себе; верификатор – сущность, которая проверяет действительность утверждения о владельце удостоверений. В представленной архитектуре стоит отметить, что верификатор и владелец удостоверений доверяют эмитенту, но не доверяют друг-другу.

Выделяют три подхода аутентификации и авторизации на основе удостоверений: с полным раскрытием атрибутов; с селективным раскрытием атрибутов; на основе доказательств с нулевым разглашением.

Основное различие между этими подходами заключается в методе, которым владелец удостоверения подтверждает утверждение о себе верификатору.

Подход, основанный на полном раскрытии атрибутов, отправляет всё удостоверения целиком верификатору в открытом виде [6]. Этот подход является самым небезопасным, так как верификатор получает всю информацию о владельце удостоверения, которая находится внутри отправляемого удостоверения.

Подход с селективным раскрытием атрибутов позволяет раскрыть только те значения атрибутов, которые необходимы для процесса верификации [7–9]. Селективное раскрытие атрибутов позволяет увеличить количество нераскрытой информации из удостоверения, таким образом данный подход предоставляет более высокий уровень безопасности данных, чем подход, основанный на полном раскрытие атрибутов.

Подход на основе удостоверений с нулевым разглашением позволяет владельцу удостоверений подтвердить утверждение о себе, не раскрывая никаких атрибутов из удостоверения [10]. В данном случае верификатор получает криптографическое доказательство, верифицировав которое он может убедиться, что утверждение о владельце удостоверении является истинным или ложным, при этом атрибуты удостоверения остаются нераскрыты. Этот подход является наиболее безопасным с точки зрения раскрытия данных владельца удостоверений. На сегодняшний день для реализации данного подхода используются неинтерактивные системы доказательств с нулевым разглашением типа zk-SNARK [11] или zk-STARK [12]. Системы подобного типа подчиняются модели, где существуют: подтверждающая сторона – сущность, которая создает доказательство, для подтверждения утверждения; проверяющая сторона – сущность, которая проверяет валидность доказательства и убеждается в истинности утверждения; система ограничений – набор логических и арифметических операций, на основе который создается и проверяется доказательство; публичные входные данные – данные, которые не-

обходимы подтверждающей и проверяющей стороне, чтобы создать и проверить доказательство; приватные входные данные – данные, которые необходимы подтверждающей стороне чтобы создать доказательство, как следует из названия, эти данные остаются нераскрытыми для проверяющей стороны.

Взаимодействие представленных компонентов происходит следующим образом:

1. Подтверждающая сторона берет систему ограничений, публичные входные данные, приватные входные данные и создает доказательство с нулевым разглашением.
2. Подтверждающая сторона отправляет созданное доказательство и публичные входные данные проверяющей стороне.
3. Проверяющая сторона на основе полученного доказательства и публичных входных данные верифицирует доказательство.

Представленная модель является общим описанием характерным для неинтерактивных систем доказательств с нулевым разглашением. В частных случаях данная модель может иметь дополнительные стадии, например стадия доверительной установки – генерация специальных ключей для создания и подтверждения доказательства, которая характерна для системы Groth-16 [13]. Каждая неинтерактивная система доказательств с нулевым разглашением имеет свой порядок слова – максимально возможное число, которым может оперировать система ограничений в базовой математической или логической операции.

4. Модель удостоверений с нулевым разглашением

Для решения проблемы унификации и независимости от конкретной платформы и системы доказательств с нулевым разглашением данная работа предлагает модель цифрового удостоверения, которое может быть использована с любой неинтерактивной системой доказательств с нулевым разглашением для аутентификации и авторизации пользователя.

Модель удостоверения с нулевым разглашением удовлетворяет следующим требованиям: идентификатором владельца удостоверения и эмитента является публичный ключ; удостоверение верифицируется в любой неинтерактивной системе доказательств с нулевым

разглашением типа zk-SNARK и zk-STARK; проверка, что удостоверение выпущено доверительным эмитентом осуществляется внутри системы ограничений.

Модель удостоверения с нулевым разглашением состоит из двух основных свойств: атрибуты (attributes) – содержат информацию о свойствах владельца удостоверений; доказательства (proofs) – содержат криптографические доказательства, которые позволяют внутри системы ограничений подтвердить, что удостоверение было выпущено доверенным эмитентом.

В свойстве атрибутов должна содержаться информация: тип удостоверения; дата выпуска удостоверения; дата начала действия удостоверения; дата окончания действия удостоверения; идентификатор владельца удостоверения – публичный ключ или блокчейн адрес, и тип публичного ключа или блокчейн адреса. Остальные свойства атрибутов зависят типа удостоверения и необходимости добавлять эти свойства для соответствия указанному типу удостоверения.

В свойстве доказательства должно содержаться как минимум одно доказательство, которое может быть проверено внутри системы ограничений и подтвердить, что удостоверение было выпущено доверенным эмитентом. В качестве такого доказательства может быть доказательство подписи эмитента. Доказательство подписи должно содержать: тип доказательства; идентификатор эмитента удостоверения – публичный ключ или блокчейн адрес, и тип публичного ключа или блокчейн адреса; подпись эмитента, которую можно верифицировать внутри системы ограничений, используя атрибуты удостоверения и идентификатор эмитента; схема преобразований атрибутов удостоверения, типа доказательства подписи, идентификатора эмитента и подписи. Схема преобразований необходима, чтобы привести каждый из атрибутов и свойств доказательства подписи в формат, который использует определенная система доказательств с нулевым разглашением в качестве приватных или публичных входных данных. Более формальное описание модели удостоверений с нулевым разглашением описано в спецификации [14].

Разработанная модель удостоверения с нулевым разглашение позволяет проводить верификацию удостоверения внутри системы ограничений используя любую неинтерактивную систему доказа-

тельств с нулевым разглашением типа zk-SNARK и zk-STARK. Процесс верификации удостоверения должен происходить согласно следующим этапам:

1. Владелец удостоверений берет необходимое удостоверение и публичные входные данные.
2. Атрибуты и доказательства удостоверения преобразуются в формат, совместимый с определенной системой доказательств с нулевым разглашением.
3. На основе системы ограничений, где происходит проверка доказательства удостоверения, что удостоверение выпущено доверительным эмитентом, и проверка соответствия атрибутов удостоверения определенным требованием, создается доказательство с нулевым разглашением.
4. Доказательство и публичные входные данные отправляются верификатору.
5. Верификатор верифицирует доказательство на основе системы ограничений и публичных входных данных, убеждаясь в правдивости или ложности утверждения.

Описанный выше процесс верификации представлен на рис. 1.

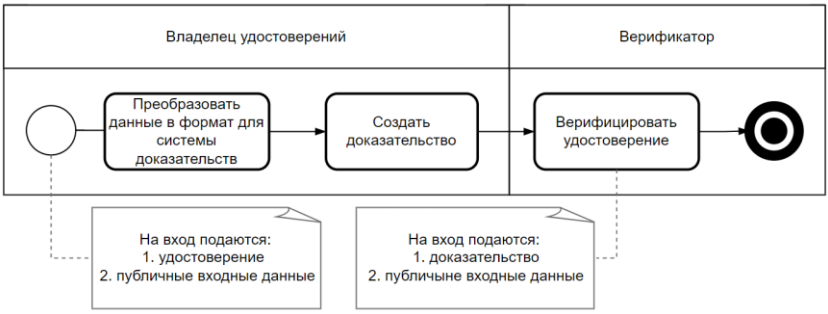


Рис. 1. Процесс верификации на основе удостоверений с нулевым разглашением

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Отчет об исследовании утечек информации ограниченного доступа в I половине 2022 года [Электронный ресурс] // INFOWATCH: разработчик решений для обеспечения информационной безопасности. – URL: https://www.infowatch.ru/sites/default/files/analytics/files/otchyot-ob-utechkakh-dannykh-za-1-polugodie-2022-goda_1.pdf (дата обращения: 25.07.2024).

2. Утечки информации ограниченного доступа в мире и в России, первое полугодие 2023 г. [Электронный ресурс] // INFOWATCH: разработчик решений для обеспечения информационной безопасности. – URL: https://www.infowatch.ru/sites/default/files/analytics/files/utechki-informatsii-ogranichennogo-dostupa-v-mire-i-rossii-za-pervoe-polugodie-2023-goda.pdf (дата обращения: 25.07.2024).
3. *Кленин Д.В*., *Максимова Е.А.* Модель вторжений в информационную систему // НБИ технологии. – 2018. – Т. 12, № 3. – С. 19-23.
4. *Максимова Е.А*., *Баранов В.В., Садовникова Н.П.* Оценка инфраструктурных рисков деструктивного характера на субъекте критической информационной инфраструктуры // Системный синтез и прикладная синергетика: Сборник научных работ X Всероссийской научной конференции, пос. Нижний Архыз, 28 сентября – 02 2021 года. – Ростов-на-Дону, Таганрог: ЮФУ, 2021. – С. 164-169.
5. *Preukschat A., Reed D*. Self-Sovereign Identity: Decentralized digital identity and verifiable credentials. – Simon and Schuster, 2021. – 504 p.
6. Verifiable credentials Data Model v2.0 [Электронный ресурс] // The World Web Wibe Consortium (W3C). – URL: https://www.w3.org/TR/vc-data-model-2.0 (дата обращения: 05.02.24).
7. The BBS Signature Scheme [Электронный ресурс] // Identity Foundation DIF. – URL: https://identity.foundation/bbs-signature/draft-irtf-cfrg-bbs-signatures.html (дата обращения: 25.07.2024).
8. Selective Disclosure for JWTs (SD-JWT) [Электронный ресурс] // IETF Datatracker. – URL: https://datatracker.ietf.org/doc/html/draft-ietf-oauth-selective-disclosure-jwt (дата обращения: 25.07.2024).
9. Merkle Disclosure Proof 2021 [Электронный ресурс] // W3C Credentials Community Group. – URL: https://w3c-ccg.github.io/Merkle-Disclosure-2021 (дата обращения: 25.07.2024).
10. *Constantinides T., Cartlidge J.P.* BlockVerify: PrivacyPreserving Zero-Knowledge Credentials Verification Framework on Ethereum // 35th European Modeling & Simulation Symposium, Greece, Athens, 18–20 Sept 2023. – Caltek, 2023. – P. 1-9.
11. *Pinto A.* An Introduction to the Use of zk-SNARKs in Blockchains // Mathematical Research for Blockchain Economys: 1st Internation Conference MARBLE. – Greece, Santorini, 2019. – P. 233-249.
12. *Tran A.M.* Theoretical and practical introduction to ZK-SNARKs and ZK-STARKs. – Masaryk University, 2022. – 146 p.
13. *Groth J*. On the Size of Pairing-based Non-interactive Arguments. – University College London, 2016. – 25 p.
14. Спецификация удостоверений с нулевым разглашением, ZCIP-2 // Github. – URL: https://github.com/zcred-org/ZCIPs (дата обращения: 25.07.2024).

УДК 004.056

М.Н. Жукова

Россия, г. Красноярск, Сибирский государственный университет науки и технологий им. академика М.Ф. Решетнева

ПРИМЕНЕНИЕ РЕЧЕВОЙ АНАЛИТИКИ ДЛЯ ДЕТЕКТИРОВАНИЯ АТАК СОЦИАЛЬНОЙ ИНЖЕНЕРИИ

В данной статье рассматриваются машинное обучение и речевая аналитика как механизмы классификации, осуществляется выбор языка программирования, описываются типы данных, компрометацию которых возможно обнаруживать, принцип работы программы и алгоритм выявления претекстинга. В рамках модуля синтаксического анализа текста разрабатываются синтаксические правила и определяются маркеры компрометации. Для определения маркеров компрометации рассматриваются основные принципы ведения злоумышленниками диалогов, специфика психологии человека и особенности речевого процесса.

Ключевые слова: *речевая аналитика; социальная инженерия; претекстинг.*

This article discusses machine learning and speech analytics as classification mechanisms, selects a programming language, describes the types of data whose compromise can be detected, the principle of the program and the algorithm for detecting pretexting. As part of the text parsing module, syntactic rules are developed and markers of compromise are identified. To identify markers of compromise, the basic principles of how attackers conduct dialogues, the specifics of human psychology and features of the speech process are considered.

Keywords: *speech analytics; social engineering; pre-texting.*

По данным Positive Technologies социальная инженерия является наиболее популярным методом атак на частных лиц и вторым по популярности в атаках на организации в 2021 году: 88% и 50% атак на частных лиц и на организации соответственно были направлены на использование человеческого фактора. [1] По мнению социального инженера ООО «Акстел-Безопасность» самые действенные методы социальной инженерии – это претекстинг и фишинг [2].

Под претекстингом понимают целевую атаку социальной инженерии, отработанную по заранее подготовленному сценарию, в результате которой субъектом конфиденциальных данных может быть осуществлена передача конфиденциальной информации злоумышленнику. Данный вид атак может осуществляться, как с применением голосовых средств (телефон, Zoom, Skype и т.д.), так и с использованием текстовых средств связи (посредством мессенджеров).

Основным способом защиты от атак с использованием техник социальной инженерии является повышение осведомленности сотрудников. Для противодействия фишингу также существуют средства защиты информации, принцип работы которых состоит в оповещении пользователя о том, что он попал на подложный или подозрительный сайт. [3] Средств защиты информации от претекстинга не существует. Задачей является разработка алгоритма выявления атак социальной инженерии типа претекстинг.

Описание основного принципа работы программного модуля

В рамках распознавания речи необходимо использовать ASR (Automatic Speech Recognition) python-библиотеку.

Системы распознавания речи имеют следующие основные модули:

а) акустическая модель;

б) языковая модель;

в) декодер [4].

Акустическая модель – это функция, принимающая на вход небольшой участок акустического сигнала (кадр или frame) и выдающая распределение вероятностей различных фонем на этом кадре. Фонема – элементарная единица человеческой речи. Таким образом, акустическая модель дает возможность по звуку восстановить, что было произнесено – с той или иной степенью уверенности [4].

Языковая модель позволяет узнать, какие последовательности слов в языке более вероятны, а какие менее. Здесь в самом простом случае требуется предсказать следующее слово по известным предыдущим словам. В традиционных системах применялись модели типа N-грамм, в которых на основе большого количества текстов оценивались распределения вероятности появления слова в зависимости от N предшествующих слов. Для получения надежных оценок распределений параметр N должен быть достаточно мал: одно, два или три слова [4].

Декодер объединяет данные от акустической и языковой моделей и преобразует их в текст с наиболее вероятной последовательностью слов [5].

Для выявления компрометации данных необходимо использовать синтаксический анализатор.

Синтаксический анализатор должен распознать структуру предложения, а именно синтаксические зависимости слов. В результате должно быть либо построено синтаксическое дерево, либо выявлены составляющие. Обычно грамматика строится так, чтобы на выходе получалось синтаксическое дерево, позволяющее выполнять разнообразные трансформации лексического содержания с пересогласованием зависимых слов, а также легко выделять семантику [6].

В рамках поставленной задачи нужно разработать правила синтаксического анализа, которые способны выявлять попытки компрометации данных. На рис. 1 отображен принцип работы программы выявления атак типа претекстинг.

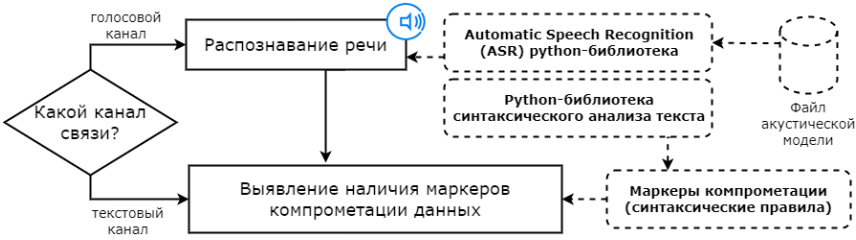


Рис. 1. Принцип работы программы выявления атак типа претекстинг

В рамках фиксации информации о каждом абоненте регистрируется информация, которая может помочь при расследовании инцидента. Например, в программу может быть включен телефонный справочник, и в карточке инцидента в соответствие номеру будут отображаться, фамилия, имя, отчество, подразделение, должность совершившего нарушение или ставшего жертвой атаки сотрудника.

Что касается действий программы при выявлении атаки, то она должна поддерживать функционал как предотвращения, так и обнаружения вторжений. В рамках обнаружения может осуществляться отправка логов в SIEM или уведомление на электронную почту, а также посредством СМС сообщения.

В общем виде, схему разработанного программного решения можно представить в виде следующей схемы (рис. 2).

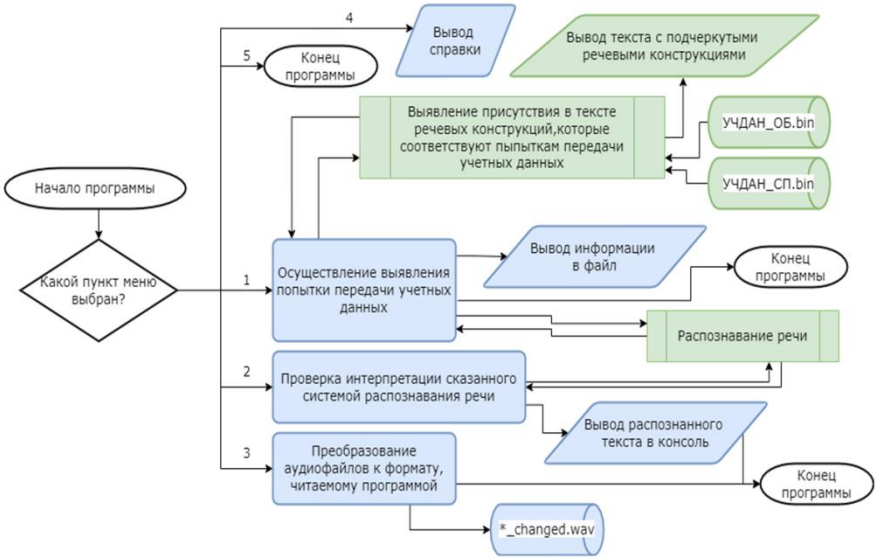


Рис. 2. Схема работы программного решения

Проведено тестирование разработанного алгоритма и программного решения. Для тестирования выбраны 4 правила для диалогов с учетом различных сценариев. Все диалоги сгенерированы и записаны в виде аудио файлов. Также записаны диалоги с внесением дополнительных шумовых эффектов, дефектов (картавость, гнусавость, заикание). В тесте участвовали 20 диалогов. Результаты распознавания и детектирования атаки представлены на рис. 3.

Полученные результаты показывают полную применимость данного подхода для решения задачи детектирования атаки типа претекстинг, совершаемой на сотрудника предприятия посредством звонка через мобильное устройство или через стационарный телефон предприятия. Разработанное решение не нуждается в встраивании в канал связи, так как позволяет путем анализа речи определять и детектировать индикаторы, позволяющие определять попытки компроментации учетных или других чувствительных данных.

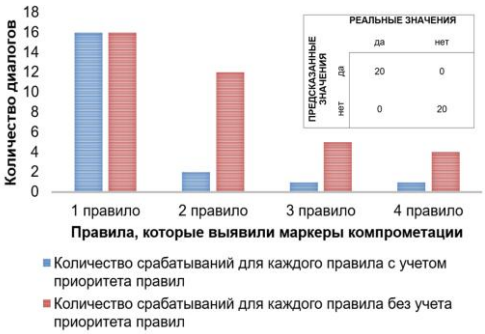


Рис. 3. Схема работы программного решения

Заключение

В рамках данной работы разработан алгоритм выявления претекстинга. В качестве классификатора необходимо использовать речевую аналитику. Программа сможет выявлять компрометацию учетных данных или коммерческой тайны. Учитывая, что аналогов данному программному решению нет, его разработка и встраивание в канал связи способно привести к повышению уровня защищенности субъектов конфиденциальных данных.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Актуальные киберугрозы: итоги 2021 года [Электронный ресурс]. – URL: https://www.ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2021/ (дата обращения: 15.07.2024).
2. Интервью социального инженера: как «взломать» человека [Электронный ресурс]. – URL: https://www.securitylab.ru/blog/company/axxtel/349562.php?ref=123 (дата обращения: 15.07.2024).
3. Антифишинг [Электронный ресурс]. – URL: https://www.anti-malware.ru/security/personal-antiphishing (дата обращения: 15.07.2024).
4. Распознавание речи [Электронный ресурс]. – URL: https://roi4cio.com/categories/category/raspoznavanie-rechi/ (дата обращения: 15.07.2024).
5. Речевые технологии. Часть 2. Speech-to-Text: как работает распознавание речи [Электронный ресурс]. – URL: https://voximplant.ru/blog/how_does_speech_to_text_work (дата обращения: 15.07.2024).
6. Синтаксический анализатор грамматического словаря [Электронный ресурс]. – URL: http://www.solarix.ru/for_developers/docs/rules.shtml (дата обращения: 15.07.2024).

УДК 004.056.5

А.В. Иванов, И.А. Огнев, В.В. Селифанов

Россия, г. Новосибирск, Новосибирский государственный технический университет

НЕКОТОРЫЕ ВОПРОСЫ ФОРМИРОВАНИЯ СВИДЕТЕЛЬСТВ ДОВЕРИЯ К ПРОЦЕССУ АУДИТА ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Для формирования доверия к процессу аудита информационной безопасности встает ряд вопросов, касающихся определения состава свидетельств или доказательств доверия, их оценки и интерпретации результатов оценки. В настоящем исследовании проводился подбор потенциальных доказательств доверия для применения их в процессе оценки доверия к процессу информационного обмена. В результате был сформирован набор свидетельств доверия, включающих в себя доказательства из разряда нормативно-правового сопровождения аудита информационной безопасности, компетенций команды аудита, полноты, качества и своевременности аудита.

Ключевые слова: *аудит; аудит информационной безопасности; оценка аудита; оценка доверия; доказательства доверия; доверенное взаимодействие; информационная безопасность; кибербезопасность.*

To build trust in the information security audit process, a number of issues arise regarding the definition of the composition of evidence or proofs of trust, their assessment and interpretation of the assessment results. In this study, a selection of potential evidence of trust was carried out for their use in the process of assessing trust in the information exchange process. As a result, a set of evidence of trust was formed, including evidence from the category of regulatory and legal support for information security audit, audit team competencies, completeness, quality and timeliness of the audit.

Keywords: *audit; information security audit; audit assessment; trust assessment; evidence of trust; trusted interaction; information security; cybersecurity.*

На сегодняшний день распространенными методами оценки процесса аудита информационной безопасности являются процессы оценки зрелости и верификации результатов аудита. Аудит инфор-

мационной безопасности – процесс получения и оценки свидетельств для подтверждения соответствия объекта аудита критериям аудита. Оба приведенных метода оценки аудита имеют ряд недостатков, заключающихся в высоких временных и трудовых затратах на реализацию оценки, а также низкой возможности автоматизации. Новый разрабатываемый метод для оценки аудита – оценка доверия к процессу аудита информационной безопасности [1, 2]. Такой метод является перспективным с точки зрения скорости и объективности оценки процесса аудита.

Целью настоящего исследования является определение набор доказательств доверия для проведения оценки доверия к процессу аудита информационной безопасности. Для достижения цели предлагается рассмотреть ряд свойств процесса аудита для выделения доказательств доверия: нормативно-правовое сопровождение, компетенции команды аудита, полнота, качество, своевременность аудита, а также поставщик услуг.

Доказательства из нормативно-правового сопровождения и качества аудита

Согласно исследованию [3] эффективность процессов аудита информационной безопасности организаций предлагается измерять при помощи показателя CSA (Control Self-Assessment). Данный показатель включает в себя 3 основные группы измеряемых аспектов аудита: планирование (Planning), исполнение (Perfoming), отчетность (Reporting).

Концептуально процедура аудита описана в различных нормативно-правовых [4, 5] и научных [6–8] источниках, и сводится к следующей последовательности действий: планирование мероприятий по обеспечению безопасности информации; формирование команды аудита; проведение предварительного обследования объекта; формирование программы аудита; сбор свидетельств аудита; оценка свидетельств аудита; выявление нарушений в реализации мер по защите информации или в организационно-распорядительной документации (ОРД); формирование отчета.

Проводя аналогию между оцениваемыми аспектами аудита и алгоритмом проведения аудита, можно сгруппировать части алгоритма следующим образом:

1) планирование
   a. формирование команды аудита;
   b. проведение предварительного обследования объекта;
   c. формирование программы аудита;
2) исполнение
   a. сбор свидетельств аудита;
   b. оценка свидетельств аудита;
   c. выявление нарушений в реализации мер по защите информации или в ОРД;
3) отчетность
   a. формирование отчета.

Исходя из вышеуказанных нормативно-правовых и научных источников [4–8] процесс аудита сопровождается рядом документов, подтверждающих реализацию того или иного этапа аудита:

1) план мероприятий по обеспечению безопасности информации;
2) план мероприятий по актуализации состава информационной системы и (или) подсистемы безопасности;
3) приказ о назначении комиссии по аудиту;
4) приказ о формировании внутреннего подразделения по аудиту;
5) соглашение или договор о предоставлении услуг по аудиту;
6) информация об активах, оцениваемых в рамках проведения аудита;
7) сведения о составе информационной системы и подсистемы защиты информации;
8) программа аудита;
9) техническое задание на аудит;
10) протоколы выявления свидетельств аудита;
11) протоколы комиссии аудита о результатах оценки свидетельств аудита;
12) заключение аудита;
13) техническое задание (частное техническое задание) на создание или совершенствование системы (подсистемы) защиты информации.

Наименования указанных документов могут отличаться в силу соблюдения терминологии нормативно-правовых актов, содержащих требования или рекомендации по обеспечению безопасности информации, в силу соблюдения профессиональной терминологии (например, при проведении активного аудита также известного как оценка защищенности или тестирование на проникновение).

Также в показателях CSA внутри критерия «Исполнение» имеется оценка использования в процессе аудиторской деятельности специализированных стандартов или фреймворков. Здесь факт использования стандартов или фреймворков свидетельствует о качестве процесса аудита как фактор учета лучших практик. Доказательством доверия в данном случае может служить положение или регламент об аудиторской деятельности внутри организации или техническое задание на аудит для внешнего поставщика услуг.

Доказательства из компетенций команды аудита

Также дополнительно стоит принять во внимание не только нормативно-правовое сопровождение, но и компетенции команды аудита [9]. Ряд публикаций предлагает проводить оценку компетенций команды аудита, рассматривая ряд метрик [10, 11]:

1. Общее количество сотрудников в команде аудита – в среднем 5 человек.
2. Количество сотрудников с сертификатом CISA – в среднем 1,4 человек.
3. Количество сотрудников с сертификатом ISO 27000 – в среднем 1,9 человек.
4. Количество сотрудников со степенью в области компьютерных наук – в среднем 1,18 человек.
5. Средний опыт работы сотрудников в команде аудита – в среднем 1,5 года для специалистов и 15 лет для руководителей.
6. Количество используемых фреймворков – в среднем 4 штук.
7. Часы дополнительного образования – в среднем 44 часа по аудиту и 10 часов по ИТ в год.

Таким образом данные метрики возможно свести к следующим доказательствам доверия:

1. Выписки из личных дел сотрудников, входящих в состав команды аудита – данное доказательство может быть использовано для анализа опыта аудиторов, их достижений, образования и курсов повышения квалификации, курсов профессиональной переподготовки и проч.
2. Положение / регламент об аудиторской деятельности внутри организации – данное доказательство позволит проанализировать количество используемых фреймворков и их актуальность.

Доказательства полноты

В исследовании [12] за основу определения эффективности тестирования на проникновение (инструментального аудита) взяты 3 показателя: полнота (достаточность свидетельств аудита для задач аудита), оперативность (время достижения задач аудита) и достоверность (точность и репрезентативность результатов аудита).

Полнота аудита может рассчитываться относительно:

1) количества проверяемых требований к системе защиты информации;
2) количества проверяемых свойств информационной безопасности (конфиденциальность, целостность, доступность);
3) количества элементов (подсистем) исследуемого объекта;
4) количества элементов (подсистем) по свойствам информационной безопасности;
5) количества уязвимостей объекта по свойствам информационной безопасности;
6) количества уязвимостей элементов по свойствам информационной безопасности;
7) количества выявленного и потенциального предотвращенного ущерба в отношении объекта, владельца, уязвимостей или свойств информационной безопасности;
8) достаточности тестового набора данных (отношение успешных информационно-технологических воздействий (ИТВ) к общему количеству ИТВ).

Данные показатели возможно вычислить, исходя из информации, содержащейся в:

1) программе аудита – планируемые к проверке элементы системы, содержания этапов аудита и сроки их исполнения;
2) техническом задании (частном техническом задании) на создание или совершенствование системы (подсистемы) защиты информации – общее количество требований по ИБ, предъявляемых к субъекту информационного обмена;
3) протоколах комиссии аудита о результатах оценки свидетельств аудита – исследованные элементы системы, затраченное время на анализ, результаты проверок;
4) заключении аудита – выявленные нарушения в системе защиты информации и системе управления информационной безопасностью (количество успешных проверок свидетельств).

Доказательства своевременности

Требования к обязательности аудита выставляются в обязательном порядке для:

1) объектов критической информационной инфраструктуры – обязательное проведение госконтроля не реже раза в 3 года и внутренней контроля или внешней оценки не реже раза в 3 года [13].
2) информационных систем персональных данных – обязательный контроль соблюдения требований по защите персональных данных своими силами или с привлечением лицензиатов ФСТЭК России не реже 1 раза в 3 года [14].
3) государственных или муниципальных информационных систем – обязательный контроль (внутренний или внешний) защищенности информации в ГИС (МИС) не реже 1 раза в год для систем 1 класса защищенности и не реже 1 раза в 2 года для систем 2 и 3 классов защищенности [15].
4) Банковских систем – обязательная самооценка или внешний аудит не реже 1 раза в 2 года [16].
5) Кредитных финансовых организаций – тестирование на проникновение не реже 1 раза в год [17].

6) Некредитных финансовых организаций – оценка уровня защиты информации не реже 1 раза в год для усиленного уровня защиты информации и не реже 1 раза в 3 года для стандартного уровня ЗИ [18].

Среди научных источников также имеется практика проведения аудита в среднем 1 раз в год или чаще [19; 20].

Доказательства доверия относительно проверки своевременности аудита информационной безопасности достаточно просты:

1. План мероприятий по обеспечению безопасности информации – задает периодичность проведения аудита.
2. Программа аудита и Заключение аудита – подтверждают факт проведения аудита в указанные в п.1 сроки.

Дополнительные доказательства доверия

Рассмотрим ряд специфических особенностей изучаемого процесса аудита информационной безопасности.

В рамках проведения аудита информационной безопасности, попадающего под требования законодательства РФ [21], или активного аудита [6] необходимо проводить анализ уязвимостей исследуемой организации, для чего необходимо применение средств анализа защищенности, сертифицированных по требованиям безопасности ФСТЭК России. Для подтверждения факта наличия таких средств необходимо доказательство доверия «Акт приема-передачи на средства контроля защищенности».

Особым рассматриваемым случаем является привлечение внешнего поставщика услуг для проведения процесса аудита информационной безопасности. Для таких поставщиков применимы все вышеперечисленные доказательства доверия, однако необходимо удостовериться дополнительно в двух вопросах:

1. Наличие у внешнего поставщика услуг лицензии ФСТЭК России на предоставление услуг из ряда лицензируемых видов деятельности (пункт б или д Постановления Правительства РФ №79) [22].
2. Добросовестность поставщика услуг – дела Арбитражных судов в отношении поставщиков услуг аудита в связи с невыполнением или недобросовестным выполнением обязательств.

Общий список доказательств доверия

Соединив доказательства доверия исходящих от нормативно-правового сопровождения аудита и компетенций команды аудита, все доказательства доверия можно привести к указанным в таблице ниже (табл. 1).

Таблица 1

Доказательства доверия

№ п/п	Наименование доказательства
1	План мероприятий по обеспечению безопасности информации
2	План мероприятий по актуализации состава информационной системы и (или) подсистемы безопасности
3	Приказ о назначении комиссии по аудиту
	Приказ о формировании внутреннего подразделения по аудиту
	Соглашение или договор о предоставлении услуг по аудиту
4	Информация об активах, оцениваемых в рамках проведения аудита
5	Сведения о составе информационной системы и подсистемы защиты информации
6	Программа аудита
	Техническое задание на аудит
7	Протоколы выявления свидетельств аудита
8	Протоколы комиссии аудита о результатах оценки свидетельств аудита
9	Заключение аудита
10	Выписки из личных дел сотрудников, входящих в состав команды аудита
11	Положение / регламент об аудиторской деятельности внутри организации
12	Техническое задание (частное техническое задание) на создание или совершенствование системы (подсистемы) защиты информации
Только для внешнего поставщика услуг	
13	Дела Арбитражных судов в отношении поставщиков услуг аудита в связи с невыполнением или недобросовестным выполнением обязательств
Только для лицензируемых видов деятельности	
14	Лицензия ФСТЭК России на предоставление услуг из ряда лицензируемых видов деятельности (пункт б или д Постановления Правительства РФ №79)
15	Акт приема-передачи на средства контроля защищенности

Приведенные доказательства доверия к аудиту могут сформировать полную картину о процессе аудита информационной безопасности изучаемой организации в силу того, что данные доказательства включают все нормативно-правовое сопровождение аудита, сведения о компетенциях команды аудита, сведения о применяемых методологиях и инструментах (в том числе программных инструментах), а также сведения об обязательных условиях поставки услуг аудита (для внешних поставщиков услуг).

Оценка доказательств доверия

Оценка доказательств доверия должна формировать исчисляемые показатели доверия ко всем аспектам аудита, поэтому помимо наличия доказательств порой необходимо проводить дополнительно оценку содержимого доказательства доверия. Разберем доказательства доверия, для которых необходимо проводить дополнительную оценку для получения дополнительных сведений (табл. 2).

Таблица 2

Компоненты доказательств доверия

Наименование доказательства	Компоненты
План мероприятий по обеспечению безопасности информации	Сведения о периодичности аудита
	Сведения об устранении нарушений, выявленных в процессе аудита
Программа аудита	Количество проверяемых требований по защите информации, подсистем или элементов информационных систем, свойств информации
Техническое задание на аудит	
Заключение аудита	Нарушения, выявленные в процессе аудита
Выписки из личных дел сотрудников, входящих в состав команды аудита	Сведения об образовании членов команды аудита
	Сведения о коммерческом опыте
	Сведения о курсах повышения квалификации
Положение / регламент об аудиторской деятельности внутри организации	Наличие ссылок на используемые стандарты, фреймворки по аудиту

Окончание табл. 2

Наименование доказательства	Компоненты
Техническое задание (частное техническое задание) на создание или совершенствование системы (подсистемы) защиты информации	Количество реализованных требований по защите информации, защищаемых подсистем или элементов информационных систем, свойств информации
Дела Арбитражных судов в отношении поставщиков услуг аудита в связи с невыполнением или недобросовестным выполнением обязательств	Проигранные дела

Приведенные выше дополнительные компоненты доказательств доверия позволят сформировать дополнительные численные параметры для оценки доверия к процессу аудита информационной безопасности. Численные показатели в данном случае являются преимуществом предлагаемой системы оценки доверия, которое заключается в возможности практически полной автоматизации процесса оценки доверия.

Заключение

В результате проведенного исследования был выделен ряд доказательств доверия (14 шт.) на основе анализа определенных свойств аудита: нормативно-правовое сопровождение аудита; компетенции команды аудита; полнота аудита; своевременность аудита; качество аудита; поставщик услуг.

Сформированные доказательства доверия опираются на документацию, касающуюся процесса аудита и команды аудита. Данный факт является строгим определением ожидаемых результатов от оценки доказательств доверия и ожидаемых документов и информации в них. Это позволяет разработать программный комплекс обработки доказательств доверия и их автоматизированную оценку, что является преимуществом по ряду причин:

1. Повышение скорости оценки доверия – максимизируя обработку доказательств доверия возможно приблизить время расчета уровня доверия к максимальному возможному времени, которое зависит лишь от вычислительных мощностей оборудования.

2. Снижение количества ошибок – строгое определение доказательств доверия и их оценки снижают количество возможных разночтений в анализе доказательств доверия, а также минимизирует участие человека в процессе оценки доверия.
3. Наличие инструмента обратной связи – высокая скорость обработки с минимальным участием человека в процессе оценки доверия позволяет использовать инструмент оценки доверия как средство самодиагностики процесса аудита ИБ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Шелупанов А.А. [и др.].* Технологии доверенного взаимодействия в экосистеме Национальной технологической инициативы // Современное образование: интеграция образования, науки, бизнеса и власти. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2022. – С. 15-20.
2. *Иванов А.В., Огнев И.А*. Проблемы оценки доверия к процессам аудита информационной безопасности // Вопросы кибербезопасности. – 2024. – № 3 (61). – С. 40-50.
3. *Slapničar S. [et al.].* Effectiveness of cybersecurity audit // International Journal of Accounting Information Systems. – 2022. – Vol. 44. – P. 100548.
4. ISO 19011:2018 - Guidelines for auditing management systems.
5. ГОСТ Р ИСО/МЭК 27007-2014 Информационная технология (ИТ). Методы и средства обеспечения безопасности. Руководства по аудиту систем менеджмента информационной безопасности.
6. *Макаренко С.И.* Аудит информационной безопасности: основные этапы, концептуальные основы, классификация мероприятий // Системы управления, связи и безопасности. – 2018. – № 1. – С. 1-29.
7. *Antunes M., Maximiano M., Gomes R.* A Customizable Web Platform to Manage Standards Compliance of Information Security and Cybersecurity Auditing: International Conference on ENTERprise Information Systems / ProjMAN - International Conference on Project MANagement / HCist - International Conference on Health and Social Care Information Systems and Technologies 2021 // Procedia Computer Science. – 2022. – Т. 196. – P. 36-43.
8. *Senkiv D.A.* Audit as a Means of Ensuring Information Security of Web Applications and Used Computer Systems // American Scientific Journal. – 2020. – Vol. 40, No. 2. – P. 54-57.
9. *Wu T.-H. [и др.].* IT governance and IT controls: Analysis from an internal auditing perspective // International Journal of Accounting Information Systems. – 2024. – Vol. 52. – IT governance and IT controls. – P. 100663.

10. *Héroux S., Fortin A.* The internal audit function in information technology governance: A holistic perspective // Journal of Information Systems. – 2013. – Vol. 27, No. 1. – P. 189-217.
11. *Ahmed I. [и др.].* A decision support model for assessing and prioritization of industry 5.0 cybersecurity challenges // Sustainable Manufacturing and Service Economics. – 2024. – Vol. 3. – P. 100018.
12. *Макаренко С.И.* Критерии и показатели оценки качества тестирования на проникновение // Вопросы кибербезопасности. – 2021. – Т. 43, № 3. – С. 43-57.
13. Федеральный закон от 26.07.2017 № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации».
14. Постановление Правительства РФ от 1 ноября 2012 г. N 1119 «Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных».
15. Приказ Федеральной службы по техническому и экспортному контролю от 18.02.2013 N 21 «Об утверждении Состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных».
16. СТО БР ИББС-1.0-2014 «Обеспечение информационной безопасности организаций банковской системы Российской Федерации. Общие положения».
17. Положение Банка России №683-П «Об установлении обязательных для кредитных организаций требований к обеспечению защиты информации при осуществлении банковской деятельности в целях противодействия осуществлению переводов денежных средств без согласия клиента».
18. Положение Банка России №684-П «Об установлении обязательных для некредитных финансовых организаций требований к обеспечению защиты информации при осуществлении деятельности в сфере финансовых рынков в целях противодействия осуществления незаконных финансовых операций».
19. *Slapničar S. [и др.].* A pathway model to five lines of accountability in cybersecurity governance // International Journal of Accounting Information Systems. – 2023. – Vol. 51. – P. 100642.
20. *Аль-Джанаби А.И.Х.* Система аудита и самооценки качества // Креативная экономика. – 2021. – Т. 15, № 4. – С. 1237-1252.
21. Приказ Федеральной службы по техническому и экспортному контролю от 25.12.2017 N 239 «Об утверждении Требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации».
22. Постановление Правительства РФ от 03.02.2012 N 79 «О лицензировании деятельности по технической защите конфиденциальной информации» (с изменениями и дополнениями).

УДК 004

А.С. Калинин

Россия, г. Таганрог, Южный федеральный университет

ОСНОВНЫЕ МЕТОДИКИ ОЦЕНКИ РИСКОЙ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ОБЪЕКТОВ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ

Целью исследования является определение основных нормативно-правовых актов и методик, необходимых на этапах проектирования и эксплуатации различных объектов критической инфраструктуры. В ходе исследования были изучены основные документы и регламенты, используемые в Российской Федерации и Европе, выявлены их основные черты и принцип работы.

Ключевые слова: *объекты критической информационной инфраструктуры (ОКИИ); нормативно-правовые акты; информационная безопасности.*

The purpose of the study is to identify the main regulatory legal acts and methodologies required at the stages of design and operation of various critical infrastructure facilities. In the course of the study the main documents and regulations used in the Russian Federation and Europe were studied, their main features and principle of operation were identified.

Keywords: *critical information infrastructure objects (CII); regulatory legal acts; information security.*

Увеличение производственных мощностей, общий рост компьютеризации и цифровизации многих аспектов повседневной жизни человека ведут за собой повышение рисков информационной безопасности. Наша «цифровая» жизнь во многом основана на работе различных сетей и телекоммуникационных систем, одним из элементов которых являются объекты критический информационной инфраструктуры (ОКИИ) – автоматизированные системы управления технологическими процессами (АСУ ТП), информационные системы (ГИС или ИСПДн) и информационно-телекоммуникационная сеть (ИТКС).

Объекты КИИ являются ключевым сегментом, обеспечивающим работоспособность общества, так как нарушение их функционирования, в связи с кибератаками, влечет за собой репутационные, социальные, финансовые и государственные ущербы. В связи с этим, при планировании модели безопасности ОКИИ, уделяется большое внимание разработке и внедрению нормативных актов, как внутренних, так и глобальных, интеграции в общую сеть новых программно-аппаратных решений, позволяющих снизить информационные риски, составлению политик безопасности и обучению персонала. К сожалению, не смотря на все аспекты, которые могут приняты, объекты КИИ до сих пор остаются уязвимы ко множеству кибератак.

Критическая инфраструктура подвержена широкому спектру угроз, как от внутренних, так и от внешних нарушителей. Можно выделить несколько основных:

- Кибервойна – поддерживаемые государством группировки могут использовать уязвимости в системах безопасности на критических объектах с целью промышленного шпионажа или для вывода из строя основных производственных сегментов, чтобы нарушить функционирование общей сети предприятий.
- Кибертерроризм – идейные группы злоумышленников, которые эксплуатируют известные уязвимости с целью дестабилизации общества.
- Киберпреступность – крупные хакерские группировки, задачей которых является кража персональных данных сотрудников или иной производственной информации, чтобы в дальнейшем использовать ее для шантажа или продажи.
- Инсайдерские угрозы – к данному типу можно отнести сотрудников объектов КИИ, которые имеют доступ к важной информации или критическим узлам производства. Чаще всего, они подвержены социальной инженерии или не имеют достаточной квалификации в сфере информационной безопасности, что влечет за собой повышение рисков.

Если рассматривать угрозы критической инфраструктуры подробнее, то можно заметить, что они не практически не отличаются от угроз в любой другой сети, системы или ином предприятии. Проанализировав особенности работы систем функционирования объектов КИИ, выделим основные угрозы:

- Отсутствие обновлений безопасности – обновление программного обеспечения, а вместе с ним и различных функций и политик безопасности, может привести к образованию уязвимостей в системах, которые, в дальнейшем, довольно проблематично устранить.
- Ограниченность финансирования – распределение финансовых средств на обеспечение всех требования безопасности также является важным критерием для защиты предприятий. Отсутствие программно-аппаратных средств фильтрации трафика или шифрования, недостаточное количество персонала, его низкая квалификация или низкое качество знаний может повлечь за собой серьезные угрозы.
- Несоответствие нормативно-правовым требованиям – разработка и ввод нормативных документов также является одним из обязательных этапов для обеспечения безопасности. Глобальные или внутренние акты, чаще всего, содержат в себе порядок действий, при возникновении угрозы, требования к персоналу, установленному оборудованию и т.д.
- Сложность проектирования – объекты критический инфраструктуры являются достаточно сложными, с технической зрения, предприятиями. Поэтому стоит обращать большое внимание на процессы проектирования систем внутреннего функционирования и внешних связей с другими предприятиями.
- Человеческий фактор – не смотря на всю автономность производственных процессов, обслуживающий персонал, штатные работники, также может нести риск возникновения угроз из-за неправильности использования системой или из злого умысла.

Как уже было сказано, формирование и принятие нормативно-правовых документов, регулирующих работу объектов критической инфраструктуры, является важной частью обеспечения их безопасности. Организации используют как внутренние акты, составленные исходя из имеющихся ресурсов, так и международных, регламентирующих общий порядок действий, свойственный для ОКИИ. Последние рассмотрим подробнее [2].

Отечественное законодательство в сфере безопасности критической инфраструктуры представлено довольно большим перечнем федеральных законов, приказов и постановлений правительства или федеральной службы по техническому и экспортному контролю

(ФСТЭК). Например, федеральный закон №182 от 26.06.2017 «О безопасности критической информационной инфраструктуры Российской Федерации» дает четкое представление об этапах категорирования объектов КИИ, требованиях по обеспечению их безопасности, необходимых государственный контроль и т.д., а приказ ФСТЭК № 239 от 25.12.2017 «Об утверждении требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры в Российской Федерации» устанавливает порядок требований к обеспечению безопасности в процессе создания, эксплуатации и вывода из эксплуатации ОКИИ, порядок внедрения технический мер защиты информации, а также утверждает общие требования к организационным мерам и программно-аппаратным средствам обеспечения информации [5].

Помимо этого, в России создано несколько координационных центров, которые структуризируют всю информацию об угрозах безопасности на объектах КИИ:

- ГосСОПКА – государственная система обнаружения, предупреждения и ликвидации последствий компьютерных атак – комплекс центров, которые обмениваются через общую сеть с другими ОКИИ информацией о вероятных кибератаках.
- НКЦКИ – национальный координационный центр по компьютерным инцидентам – регулятор, стоящий «над» ГосСОПКА, предназначенный для сбора и анализа информации об атаках на ОКИИ и предоставления объектам критической инфраструктуры данных для защиты от потенциального нарушения работы системы [1].

Европейское и американское законодательство, по большей части, основано на применении общих стандартов, таких как ISO/IEC 27001:2013 и NIST CSF 1.1. Оба стандарта являются результатом категорирования информации и разработки определенных методологий международного института стандартизации (NIST) и международной электротехнической комиссии [3].

Стандарт ISO/IEC 27001:2013 рассматривает все необходимые требования к различным системам как ручного, так и автоматизированного управления информационной безопасностью и также предусматривает 4 основных процесса управления: планирование, реализация, проверка, совершенствование [6].

NIST CSF 1.1, в свою очередь, является общей методологией повышения уровня защищенности критически важных объектов. Представляет собой рамочную программу, основанную на различных международных стандартах, которая предлагает выполнить алгоритм выявления рисков информационной безопасности и повышения защищенности системы. Сам же алгоритм состоит из следующих этапов:

- Идентификация – определяется ценность имеющихся активов в удобном для понимания эквиваленте, проведение инвентаризации.
- Защита – составляется перечень необходимых СЗИ (средств защиты информации), которые планируется внедрить в систему объекта критической инфраструктуры.
- Обнаружение – составляются и внедряются соответствующие регламенты и правил, позволяющих выявить вероятные события безопасности.
- Реагирование – разработка и последующее внедрение регламентов для реагирования на обнаружение инцидентов информационной безопасности.
- Восстановление – создание алгоритма действий, направленного на устранение последствий после нарушения работы системы [4, 7].

Таким образом, для специалистов информационной безопасности существует довольно большой спектр нормативно-правовых актов, стандартов и общих правил, которые необходимо использовать при проектировании и эксплуатации ОКИИ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Проблемы правового обеспечения безопасности критической информационной инфраструктуры Российской Федерации [Электронный ресурс]. – URL: https://ii.org.ru/problemy-pravovogo-obespecheniya-bezo/?ysclid=lxkiqvxgra609551959.
2. Приказ ФСТЭК РОСИИ от 25.12.2017 №239 [Электронный ресурс]. – URL: https://fstec.ru/dokumenty/vse-dokumenty/prikazy/prikaz-fstek-rossii-ot-25-dekabrya-2017-g-n-239?ysclid=lxs75njicx848581789.
3. Four cybersecurity challenges that critical infrastructures are facing [Электронный ресурс]. – URL: https://www.threatq.com/cybersecurity-challenges-critical-infrastructure/.

4. Critical infrastructures security: Challenges and best practices [Электронный ресурс]. – URL: https://www.cybertalk.org/2023/05/30/critical-infrastructure-security-challenges-and-best-practices/.
5. Обеспечение безопасности значимых объектов критической информационной инфраструктуры (КИИ) [Электронный ресурс]. – URL: https://www.ec-rs.ru/resheniya/bezopasnost-kriticheskoy-informatsionnoy-infrastruktury-kii/?ysclid=lxrktripz5176883885.
6. Обеспечение безопасности КИИ [Электронный ресурс]. – URL: https://data-sec.ru/services/critical-infrastructure/kii-protection/?ysclid=lxrkxw7tub702122281.
7. Understanding Cyber Security in Critical Infrastructure [Электронный ресурс]. – URL: https://www.geeksforgeeks.org/understanding-cyber-security-in-critical-infrastructure/.

УДК 004.052.2

И.А. Калмыков, И.Д. Ефременков, Д.В. Духовный

Россия, г. Ставрополь, Северо-Кавказский федеральный университет

ПОМЕХОУСТОЙЧИВЫЙ ПРОТОКОЛ АУТЕНТИФИКАЦИИ НИЗКООРБИТАЛЬНОГО СПУТНИКА, РЕАЛИЗУЕМЫЙ В МОДУЛЯРНОМ КОДЕ

Решить проблему доступа к ресурсам Интернета в районах Крайнего Севера можно с помощью низкоорбитальной системы спутникового интернета (НССИ). Однако при увеличении числа НССИ возрастает вероятность навязывание со стороны спутника-нарушителя неавторизованного контента. Уменьшить вероятность навязывания можно с помощью протокола аутентификации спутника (ПАС), реализованного в модулярных кодах (МК). Применение МК позволяет не только повысить скорость опознавания спутника, но корректировать ошибки, вызванные помехами. Цель – разработать алгоритм коррекции ошибок в МК, обладающий минимальными временными затратами и оказывающий меньшее негативное воздействие на скорость опознавания.

Ключевые слова: *протокол аутентификации; модулярные коды; коррекция ошибки.*

The problem of access to Internet resources in the Far North can be solved with the help of a low-orbit satellite Internet system (LSIS). However, with an increase in the number of LSIS, the likelihood of unauthorized content being imposed by the offending satellite increases. To reduce the likelihood of imposition, you can use the satellite authentication protocol (SAP), implemented in modular codes (MC). The use of the MC allows not only to increase the speed of satellite identification, but also to correct errors caused by interference. The goal is to develop an error correction algorithm in the MC that has minimal time costs and has less negative impact on the recognition speed.

Keywords: *authentication protocol; modular codes; error correction.*

Решить проблему всеобщего доступа к ресурсам сети Интернета в районах Крайнего Севера можно за счет использования низкоорбитальных систем спутникового интернета (НССИ) [1, 2]. По мере расширения числа НССИ возрастает вероятность навязывание со

стороны спутник-нарушителя неавторизованного контента. Предотвратить данную ситуацию можно за счет протокола аутентификации спутника (ПАС) перед началом сеанса связи. Если спутник получит статус «свой», то ему будет предоставлен сеанс связи. В противном случае – в сеансе связи будет отказано. Для обеспечения высокой имитостойкости в работах [3, 4] был предложен протокол аутентификации с нулевым разглашением знаний, реализованный в модулярных кодах.

В модулярных кодах целое число U представляется в виде $U = (U_1, U_2, ..., U_k)$, где $U_i \equiv U \bmod m_i$, m_i – взаимно простые целые числа, являющиеся основаниями МК, где $i = 1, ..., k$ [5–7]. Тогда для МК справедливо выражение (1)

$$U * Y = ((U_1 * Y_1) \bmod m_1, ..., (U_k * Y_k) \bmod m_k), \quad (1)$$

где * – операции сложения, вычитания и умножения; $Y = Y_i \bmod m_i$; $i = 1, ..., k$.

Произведение оснований дает рабочий диапазон

$$M_k = \prod_{i=1}^{k} m_i . \quad (2)$$

Чтобы исправить однократную ошибку (один искаженный остаток) в МК вводят два контрольных основания, которые удовлетворяют условию (3)

$$m_{k+1} m_{k+2} > m_{k-1} m_k . \quad (3)$$

В результате увеличивается диапазон и становится полным согласно (4)

$$M_{k+2} = \prod_{i=1}^{k+2} m_i = M_k m_{k+1} m_{k+2} = M_k M_2^* . \quad (4)$$

Избыточная комбинация МК не содержит ошибки, если число U не превысит рабочий диапазон, то есть выполняется условие (5)

$$U = (U_1, U_2, ..., U_k, U_{k+1}, U_{k+2}) < M_k . \quad (5)$$

Для коррекции ошибок в МК был разработан ряд алгоритмов. В работах [5, 6] для коррекции ошибок предлагается использовать алгоритм проекции. Для получения проекции необходимо из $U=(U_1,U_2,U_3,...,U_{n+2})$ последовательно удалять остатки. Так первая проекция имеет вид $\widetilde{U}^1=(U_2,U_3,....U_{n+2})$. Вторая проекция имеет вид $\widetilde{U}^2=(U_1,U_3,....U_{n+2})$. Затем каждую проекцию переводят в позиционный код, используя Китайскую теорему об остатках (КТО)

$$U=\sum_{i=1}^{k+2}U_i\,B_i\ \bmod M_{k+2}=\left|U_1^*B_1\ +...+U_g^*B_g\ +...+U_{n+2}B_{n+2}\right|_{M_{n+2}},\quad (6)$$

где B_i – ортогональные базисы; $B_i=M_{k+2}w_i\,/m_i$; $B_i=1\bmod m_i$; $w_i\left|B_i\right|_{m_i}^{+}=1\bmod m_i$.

Ошибка произошла в *g*-ом остатке, если выполняется условие (7)

$$\begin{aligned}&\left\{\widetilde{U}_1,...,\widetilde{U}_{g-1},\widetilde{U}_{g+1},...,\widetilde{U}_{n+2}\right\}>M_k,\\&\left\{\widetilde{U}_g\right\}<M_k,\end{aligned}\quad (7)$$

В работе [7] предлагается использовать позиционную характеристику (ПХ) – интервал числа, которая определяется (8)

$$S=\left[U/M_k\right],\quad (8)$$

где [*] – целая часть результата деления числа *U* на рабочий диапазон.

Если комбинация не содержит ошибок, то *S* = 0. В работе [6] описан алгоритм, использующий функцию Эйлера согласно (9)

$$S=\left|\sum_{i=1}^{k+2}\left|S_{U_i}\,U_i\right|_{M_k}^{+}\right|_{M_k}^{+},\quad (9)$$

где $S_{U_i}=\left|\dfrac{M_i^{\varphi(p_i)}}{M_{k+2}}\right|_{M_k}^{+}$; $\varphi(m_i)$ – функция Эйлера числа m_i; $M_i=\dfrac{M_{k+2}}{m_i}$.

Недостатком рассмотренных алгоритмов являются большие временные затраты на коррекцию ошибок. Устранить этот недостаток позволяет разработанный алгоритм коррекции ошибки. В основе алгоритма лежит ПХ интервал. Подставив в выражение (8) равенство (6), имеем

$$S = \left[\sum_{i=1}^{k+2} U_i \, B_i \; - r_U M_{k+2} \Big/ M_k \right], \tag{10}$$

где r_u – ранг числа U.

Если взять ортогональные базисы информационных оснований, то имеем

$$B_i = \left[B_i \Big/ M_k \right] \cdot M_k + \ddot{\ddot{B}}_i = K_i M_k + \ddot{\ddot{B}}_i \, , \tag{11}$$

где $\ddot{\ddot{B}}_i$ – ортогональные базисы МК, в составе которого используются только информационные основания; $\ddot{\ddot{B}}_i \equiv B_i \mod M_k$; $i = 1,2,\ldots,k$.

Для избыточных оснований, где $i = k+1, k+2$, имеет место равенство (12)

$$B_i = K_i M_k \, , \tag{12}$$

Так как полный диапазон согласно (4) равен $M_{k+2} = M_k M_2^*$ то в нем содержится M_2^* рабочих интервалов с номерами от 0 до $M_2^* - 1$. В этом случае выражение (10) можно свести к вычислению по модулю M_2^*. Подставим равенства (11) и (12) в выражение (10). Получаем

$$S = \left| \frac{\sum_{i=1}^{k+2} U_i \left(K_i M_k + \ddot{\ddot{B}}_i \right)}{M_k} \right|_{M_2^*}^{+} = \left| \sum_{i=1}^{k+2} U_i K_i + \left[\frac{\sum_{j=1}^{k} U_j \ddot{\ddot{B}}_j}{P_n} \right] \right|_{M_2^*}^{+} = \left| \sum_{i=1}^{k+2} U_i K_i + \ddot{\ddot{r}}_U \right|_{M_2^*}^{+}, \tag{13}$$

где $\ddot{\ddot{r}}_U$ – ранг числа U, представленного в МК с основаниями $m_1, m_2, ..., m_k$.

В качества недостатка алгоритма (13) можно выделить вычисление по составному модулю $M_2^* = m_{k+1}m_{k+2}$, что приводит к увеличению аппаратных и временных затрат при коррекции МК. Устранить этот недостаток можно с помощью разработанного алгоритма коррекции, построенного на изоморфизме КТО за счет перехода к параллельным вычислениям по контрольным основаниям

$$S_{k+1} = |S|^+_{m_{k+1}} = \left| \sum_{i=1}^{k+2} U_i \left| K_i \right|^+_{m_{k+1}} + \ddot{r}_U \right|^+_{m_{k+1}}, \; S_{k+2} = |S|^+_{m_{k+2}} = \left| \sum_{i=1}^{k+2} U_i \left| K_i \right|^+_{m_{k+2}} + \ddot{r}_U \right|^+_{m_{k+2}}, \quad (14)$$

В алгоритме (14) сокращение временных затрат достигается за счет того, что для основания m_{k+j}, где $j = 1,2$ только $K_{k+j} \neq 0$, а остальные раны нулю. Данный алгоритм можно использовать для контрольных остатков на передающей стороне. Пусть МК имеет k информационные и одно контрольное основания. Тогда выражение (14) имеет вид

$$S_{k+1} = \left| U_1 \left| K_1 \right|_{m_{k+1}} + U_2 \left| K_2 \right|_{m_{k+1}} + ... + U_{k+1} \left| K_{k+11} \right|_{m_{k+1}} + \dddot{r}_U \right|^+_{m_{k+1}}. \quad (15)$$

Избыточная комбинация МК не имеет ошибку, если $S_{k+1} = 0$.

Тогда на основе (15) контрольный остаток определяется выражением

$$U_{k+1} = m_{k+1} \left| U_1 \left| K_1 \right|_{m_{k+1}} + U_2 \left| K_2 \right|_{m_{k+1}} + ... + U_k \left| K_k \right|_{m_{k+1}} + \ddot{r}_U \right|^+_{m_{k+1}} \Big/ K_{m+1}. \quad (16)$$

В этом случае разработанный помехоустойчивый протокол аутентификации, реализованный в модулярном коде, имеет вид. На предварительном этапе выбирается кортеж информационных оснований $m_1, m_2, ..., m_k$, удовлетворяющих условию $M_k > Q$, где Q – простое число, используемое в одномодульном протоколе [19, 20]. Секретные параметры переводятся в МК: $U = (U_1, U_2, ..., U_k)$ – секретный ключ спутника; $S(j) = (S_1(j), ..., S_k(j))$ – сеансовый

ключ; $T(j) = (T_1(j), ..., T_k(j))$ – число для уравнения «повторного применения сеансового ключа».Выбираются контрольные основания m_{k+1}, m_{k+2}, удовлетворяющие условию (3).

На первом этапе протокола необходимо выполнить вычисления:

1. Претендент P (спутник) вычисляет истинный статус спутника в МК

$$C_i = \left| g^{U_1} g^{S_1(j)} g^{T_1(j)} \right|_{m_i}^{+}, \qquad (17)$$

где g – порождающий элемент по модулю m_i; i =1,2,…, k.

2. Претендент выбирает случайные числа $\Delta U,(j)\,\Delta S(j), \Delta T(j)$ для выполнения операции «зашумление» секретных параметров протокола

$$U_i^* = \left| U_i + \Delta U_i(j) \right|_{\varphi(m_i)}^{+}; \; S_i^*(j) = \left| S_i(j) + \Delta S_i(j) \right|_{\varphi(m_i)}^{+};$$

$$T_i^*(j) = \left| T_i(j) + \Delta T_i(j) \right|_{\varphi(m_i)}^{+}. \qquad (18)$$

где $\Delta U_i = \left|\Delta U(j)\right|_{m_i}^{+}, \Delta S_i = \left|\Delta S(j)\right|_{m_i}^{+}, \Delta T_i = \left|\Delta T(j)\right|_{m_i}^{+}$; $i = 1,..., k$; j – номер сеанса.

3. Претендент P вычисляет «зашумленный» статус спутника в МК

$$C_i^* = \left| g^{u_i^*} g^{S_i^*(j)} g^{T_i^*(j)} \right|_{m_i}^{+}. \qquad (19)$$

Второй этап протокола аутентификации состоит из следующих этапов.

1. Проверяющий V (абонент) передает P случайное число $d = (d_1, ..., d_{k+2})$.

2. Претендент P, получив «вопрос» $d = (d_1, d_2, ..., d_{k+2})$, проверяет его, используя алгоритм (14). А затем вычисляет ответы

$$r_i(1) = \left| U_i^* - d_i U_i \right|_{\varphi(m_i)}^{+}; \; r_i(2) = \left| S_i^*(j) - d_i S_i(j) \right|_{\varphi(m_i)}^{+};$$

$$r_i(3) = \left| T_i^*(j) - d_i T_i(j) \right|_{\varphi(m_i)}^{+}. \qquad (20)$$

Претендент с помощью алгоритма (16) вычисляет контрольные основания и генерирует сигнал, который передает проверяющему V

$$\left\{(C_1, ..., C_{k+2}), (C_1^*, ..., C_{k+2}^*), (r_1(1), ..., r_{k+2}(1)), (r_1(2), ..., r_{k+2}(2)), (r_1(3), ..., r_{k+2}(3))\right\}.$$

3. Проверяющий V с помощью алгоритма (14) корректирует ошибки в принятом сигнала, а затем выполняет проверку полученных «ответов»

$$Y_i = \left|\left(C_i\right)^{d_i} g^{r_i(1)} g^{r_i(2)} g^{r_i(3)}\right|_{m_i}^{+}. \qquad (21)$$

Претендент Р получает статус «свой», если $\left\{Y_1 = C_1^*, Y_2 = C_2^*, ..., Y_k = C_k^*\right\}$.

Результаты экспериментальных исследований

Пусть заданы информационные основания $m_1 = 11, m_2 = 13, m_3 = 19$, у которых имеется g = 2. Рабочий диапазон $M_3 = 2717$. Пусть секретные параметры равны $U = 230 = (10,9,2);\ S(j) = 1000 = (10,12,12); T(j) = 7 = (7,7,7)$. Контрольными основаниями выбираем $m_4 = 29, m_5 = 37$. На первом этапе выполняются следующие вычисления:

1. $C_1 = \left|2^{10} \cdot 2^{10} \cdot 2^7\right|_{11}^{+} = \left|2^7\right|_{11}^{+} = 7$; $C_2 = \left|2^9 \cdot 2^{12} \cdot 2^7\right|_{13}^{+} = \left|2^4\right|_{13}^{+} = 3$;

$C_3 = \left|2^2 \cdot 2^{12} \cdot 2^7\right|_{19}^{+} = 8$.

2. Выбираем $\Delta U(j) = (1,4,7);\ \Delta S(j) = (3,8,4);, \Delta T(j) = (1,1,2)$. После «зашумления» получаем параметры

$U^* = (11,1,9);\ S^*(j) = (38,16); T^*(j) = (8,8,9)$.

3. $C_1^* = \left|2^{11} \cdot 2^3 \cdot 2^8\right|_{11}^{+} = \left|2^2\right|_{11}^{+} = 4$; $C_2^* = \left|2^1 \cdot 2^8 \cdot 2^8\right|_{13}^{+} = \left|2^5\right|_{13}^{+} = 6$;

$C_3 = \left|2^9 \cdot 2^{16} \cdot 2^9\right|_{19}^{+} = \left|2^{16}\right|_{19}^{+} = 5$.

На втором этапе выполняются следующие вычисления:

1. Пусть вопрос $d(j) = 100 = (1,9,5,13,26)$, который передается на спутник.

2. Пусть принята комбинация $d'''(j) = (0,9,5,13,26)$. Претендент Р проверяет код с помощью алгоритма (14). Ортогональные базисы имеют вид

$B_1 = 1855217 = 682 \cdot M_k + 2223$; $B_2 = 4485014 = 165 \cdot M_k + 209$; $B_3 = 2301585 = 847 \cdot M_k + 286$;;

$B_4 = 201058 = 74 \cdot M_k$;; $B_5 = 1024309 = 377 \cdot M_k$.

Претендент вычисляет ранг числа $\dddot{r}_U = \left[\frac{0 \cdot 2223 + 9 \cdot 209 + 5 \cdot 286}{2717}\right] = 1$.

Тогда

$$S_4 = \left|0 \cdot 682 + \cdot 9 \cdot 165 + 5 \cdot 847 + 13 \cdot 74 + 26 \cdot 377 + 1\right|_{29}^{+} = \left|0 \cdot 15 + \cdot 9 \cdot 20 + 5 \cdot 6 + 13 \cdot 16 + 26 \cdot 0 + 1\right|_{29}^{+} = 13.$$

$$S_5 = \left|0 \cdot 682 + \cdot 9 \cdot 165 + 5 \cdot 847 + 13 \cdot 74 + 26 \cdot 377 + 1\right|_{37}^{+} = \left|0 \cdot 16 + \cdot 9 \cdot 17 + 5 \cdot 33 + 13 \cdot 0 + 26 \cdot 7 + 1\right|_{7}^{+} = 20.$$

Этой ПХ соответствует вектор ошибки $\bar{e}(j) = (1,0,0,0,0)$. Претендент корректирует ошибку

$$d(j) = d'''(j) + \bar{e}(j) = (0,9,5,13,26) + (1,0,0,0,0) = (1,9,5,13,26)$$

и вычисляет (20).

$$r_1(1) = \left|11 - 1 \cdot 10\right|_{\varphi(11)}^{+} = 11; r_2(1) = \left|1 - 9 \cdot 9\right|_{\varphi(13)}^{+} = 4; r_3(1) = \left|9 - 5 \cdot 2\right|_{\varphi(19)}^{+} = 17.$$

$$r_1(2) = \left|3 - 1 \cdot 10\right|_{10}^{+} = 3; r_2(2) = \left|8 - 9 \cdot 12\right|_{12}^{+} = 8; r_3(2) = \left|16 - 5 \cdot 12\right|_{18}^{+} = \left|-8\right|_{18}^{+} = 18 - 8 = 10.$$

$$r_1(3) = \left|8 - 1 \cdot 7\right|_{10}^{+} = 1; r_2(3) = \left|8 - 9 \cdot 7\right|_{12}^{+} = 5; r_3(3) = \left|9 - 5 \cdot 7\right|_{18}^{+} = \left|-8\right|_{18}^{+} = 18 - 8 = 10.$$

Претендент, используя (16), вычисляет контрольные остатки для сигнала

$$\left\{C(j) = (7,3,8,20,28); C^*(j) = (4,6,5,2,5); r(1) = (11,4,17,18,9); r(2) = (3,8,10,2,4); r(3) = (1,5,10,13,28)\right\}.$$

Сигнал передается проверяющему V.

3. Проверяющий сначала, используя алгоритм (14) корректирует ошибки в сигнале, а затем осуществляет проверку согласно (9)

$$Y_1 = \left|\left(C_1\right)^{d_1} g^{r_1(1)} g^{r_1(2)} g^{r_1(3)}\right|_{m_1}^{+} = \left|7^1 \cdot 2^{11} \cdot 2^3 \cdot 2^1\right|_{11}^{+} = \left|2^2\right|_{11}^{+} = 4.$$

$$Y_2 = \left|\left(C_2\right)^{d_2} g^{r_2(1)} g^{r_2(2)} g^{r_2(3)}\right|_{m_2}^{+} = \left|3^9 \cdot 2^4 \cdot 2^8 \cdot 2^5\right|_{13}^{+} = \left|2^5\right|_{13}^{+} = 6.$$

$$Y_3 = \left|\left(C_3\right)^{d_3} g^{r_3(1)} g^{r_3(2)} g^{r_3(3)}\right|_{m_1}^{+} = \left|8^5 \cdot 2^{17} \cdot 2^{10} \cdot 2^{10}\right|_{19}^{+} = \left|2^{16}\right|_{19}^{+} = 5.$$

Претендент получает статус «свой», так как

$$\left\{ Y_1 = C_1^* = 4, Y_2 = C_2^* = 6,, Y_3 = C_3^* = 5 \right\}.$$

Для оценки временных затрат на реализацию разработанного алгоритма коррекции для протокола аутентификации был использован FPGA Xilinx Artix-7 (xc7a12ticsg325-1L). При использовании алгоритма (13) время коррекции ошибки составило 152 нс. При использовании разработанного алгоритма (14) время коррекции сократилось в 1,24 раза и составило 123 нс. Таким образом, поставленная цель работы достигнута.

Исследование выполнено за счет гранта Российского научного фонда, grant number 23-21-00036, https://rscf.ru/en/project/23-21-00036/.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *McDowell J. C.* The Low Earth Orbit Satellite Population and Impacts of the SpaceX Starlink Constellation // The Astrophysical Journal Letters, 892:L36 (10 pp), 2020 April 1. – DOI: 10.3847/2041-8213/ab8016.
2. *Dang T., Li X., Luo B.* Unveiling the Space Weather During the Starlink Satellites Destruction Event on 4 February 2022 // Space Weather. – 2022. – Vol. 20, Issue 8. DOI: org/10.1029/2022SW003152.
3. *Чистоусов Н.К., Калмыков И.А., Чипига А.Ф., Калмыкова Н.И.* Разработка протоколов аутентификации низкоорбитальных космических аппаратов на основе параллельных кодов систем остаточных классов // Инженерный вестник Дона. – 2021. – № 4. – ivdon.ru/ru/magazine/archive/n4y2021/6912. (дата обращения: 16.08.2024).
4. *Olenev A.A., Kalmykov I.A., Pashintsev V.P.* Improved Spacecraft Authentication Method for Satellite Internet System Using Residue Codes / July 2023 Information 14 (7):407Follow journal. – DOI: 10.3390/info14070407.
5. *Tyncherov K.T., Mukhametshin V.Sh., Khuzina L.B.* Method to control and correct telemtry well information in the basis of residue number system // Journal of Fundamental and Applied Sciences. – 2017. – Vol. 9 (2). – P. 1370-1374.
6. *Ananda Mohan.* Residue Number Systems. Theory and Applications. – Springer International Publishing Switzerland, 2016. – 351 p.
7. *Червяков Н.И., Коляда А.А., Ляхов П.А.* Модулярная арифметика и ее приложения в инфокоммуникационных технологиях. – М.: Физматлит, 2017. – 400 с.

УДК 004.056.55

А.П. Кирьянова

Россия, г. Санкт-Петербург, Университет ИТМО

ЛОГИЧЕСКИЙ КРИПТОАНАЛИЗ ШИФРА ГОСТ Р 34.12-2015 «МАГМА»

В работе рассматривается симметричный блочный шифр «Магма». Проводится анализ его структуры, применяется логический криптоанализ для поиска ключа для разных статистических пар неполнораудовой версии шифра. В результате работы было найдено 228 бит ключа.

Ключевые слова: *шифр «Магма»; блочный шифр; логический криптоанализ; CBMC; Kissat.*

The paper considers the symmetric block cipher "Magma". Its structure is analyzed, logical cryptanalysis is used to find the key for different statistical pairs of the non-full-bit version of the cipher. As a result of the work, 228 bits of the key were found.

Keywords: *"Magma" cipher; block cipher; logical cryptanalysis; CBMC; Kissat.*

1. Введение

Шифрование – это процесс изменения сообщения таким образом, чтобы у третьих лиц не было возможности получить исходное сообщение без знания ключевой информации. Современные шифры обладают стойкостью к частотному и статистическому анализу, но с появлением новых шифров появляются и новые методы для анализа и атак.

В данной работе рассмотрен симметричный блочный шифр «Магма» (в режиме работы простой замены), на данный момент являющийся стандартом шифрования в России, изучена его структура и принцип работы, а также проанализирована его стойкость к логическому криптоанализу.

2. Блочный шифр «Магма»

2.1. Описание и принцип работы

Отечественные симметричные алгоритмы шифрования ГОСТ Р 34.12–2018 «Кузнечик» и «Магма» были разработаны Федеральной Службой Безопасности России совместно с компанией «ИнфоТеКС», первые публикации датируются 2015 годом [1]. С 2016 года данные алгоритмы являются стандартами шифрования вместо ГОСТ 28147–89, созданного в 1978 году, хотя во многом заимствуют его структуру. С 2019 года шифры «Магма» и «Кузнечик» используются в протоколах ESP и IKEv2.

Основой блочного шифра «Магма» является сеть Фейстеля, на вход принимается блок открытого текста длиной 64 бита, который итеративно обрабатывается функцией F на раундовом 32-битном ключе k_i (общая длина ключа – 256 бит) в течение 32 раундов, на выходе получается 64-битная строка шифртекста. Операциями для работы с блоками открытого текста являются циклический битовый сдвиг влево (<<<), XOR ($\oplus$), сложение по модулю 2^{32} ($\boxplus$) и перестановка π. Схема одного из 32 раундов приведена на рис. 1.

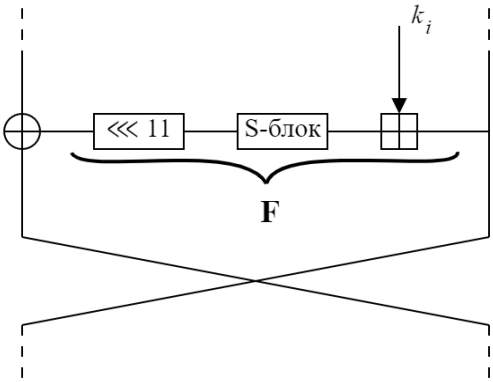


Рис. 1. Один раунд блочного шифра «Магма»

Первым шагом при работе с открытым текстом блок длиной 64 бита разбивается на две части, по 32 бита каждая. На первом раунде правая половина подаётся на вход функции F (внутри которой

изменяется перестановкой π и циклическим сдвигом), полученный результат складывается по модулю 2 с левой половиной, после чего половины меняются местами. Данный алгоритм повторяется ещё 31 раз. Алгоритм расшифрования почти полностью идентичен алгоритму шифрования за исключением того, что порядок используемых ключей инвертируется.

2.2. Криптоанализ шифра «Магма»

Одним из важных шагов при анализе безопасности любого криптографического алгоритма является проверка его стойкости к различным атакам, чем занимается криптоанализ.

Алгоритмы шифрования состоят из трёх основных частей: открытый текст, ключ (или ключи) и шифртекст. Основной задачей злоумышленника, обладающего только шифртекстами, может быть дешифровка исходной информации. Если атакующий обладает одной или несколькими статистическими парами «открытый текст–шифртекст», то его задачей может быть восстановление ключа для дальнейшего использования.

С момента публикации алгоритма шифрования «Магма» в открытых источниках было сделано опубликовано множество различных работ по криптоанализу шифра. В 2016 году Ищукова Е.А., Богданов К.И. и Бабенко Л.К. провели слайдовую атаку [2]. Слайдовая (или сдвиговая) атака – это вариация атаки на основе подобранного открытого текста (chosen-plaintext attack). Суть заключается в проведении двух параллельных шифрований, одно из которых "отстаёт" от другого на один раунд – таким образом происходит сдвиг, отсюда и название. После завершения 32 раундов происходит сравнение полученных слайдовых пар для вычисления битов раундового ключа, а позже с помощью других слайдовых пар получение 256 битов ключа.

В 2017 году Бабенко Л.К., Маро Е.А. и Аникеевым М.В. была опубликована работа по анализу стойкости «Магмы» к алгебраическому криптоанализу [3]. При таком подходе составляется система булевых уравнений. Авторы использовали среду SageMath для поиска битов ключа на известных статистических парах. После построения системы нелинейных булевых уравнений, описывающих S-блоки, сложение по модулю 2 и функцию выработки раундового ключа, было использование метода расширенной линеаризации либо нахождение выполняющего набора (битов ключа) с помощью алгоритмов SAT.

3. SAT-криптоанализ шифра «Магма»

«Магма» обладает хорошими свойствами рассеивания и запутывания, поэтому построить линейное приближение данного шифра – нетривиальная задача. На помощь может прийти логический или SAT-криптоанализ. Основная идея данного подхода заключается в преобразовании алгоритма в систему булевых уравнений и поиска выполняющего набора, при котором полученная формула принимает значение «истина».

В работе была реализована атака на основе подобранного открытого текста, таким образом криптоаналитик имеет доступ к алгоритму шифрования «Магма» и обладает несколькими открытыми и соответствующими им шифртекстами.

За основу была взята реализация шифра «Магма» на языке C++ с GitHub [4], для дальнейшей работы код был переписан на язык C: были исследованы только функция F и функция шифрования, не были задействованы алгоритм развёртки ключа и функция расшифровки. Все вычисления проводились на персональном компьютере со следующими характеристиками: ОС Ubuntu, процессор Intel i7-8550, 8 ядер, 16 Гб оперативной памяти.

Код на языке C подавался на вход программному средству CBMC (C Bounded Model Checker) для генерации булевой формулы, записанной в конъюнктивной нормальной форме (КНФ). Для получение корректного значения ключа было сгенерировано несколько КНФ на различных наборах статистических пар. Значения открытых и зашифрованных текстов были известны и зафиксированы в программе с помощью функции `CPROVER_assume()`.

Полученный файл с расширением `.cnf` подавался на вход SAT-решателю Kissat, основанному на алгоритме CDCL, для нахождения выполняющего набора.

Из решения, полученного при помощи Kissat, выделялись биты ключа и переводились в шестнадцатеричную систему счисления для сравнения с эталонным значением. Полученные результаты приведены в табл. 1.

Таблица 1

Время нахождения ключа неполнораундовой версии шифра для различных статистических пар

Количество пар текстов	Количество раундов	Количество переменных	Количество дизъюнктов	Время выполнения, чч:мм:сс
5	5	85018	368087	00:00:14
21	6	331762	1540709	09:21:03
21	8	432730	2046893	не решено

Заключение

При анализе было использовано 21 статистическая пара «открытый текст–шифртекст» при анализе 6 раундов, что позволило в итоге за почти 10 часов получить 228 корректных битов ключа:

5 раундов (160 бит):

ffeeddccbbaa9988 7766554433221100f0f1f2f3ffffffff ffffffffffffffff

6 раундов (228 бит):

ffeeddccbbaa9988 7766554433221100 f0f1f2f3f4f5f6f7 ffffffffffffffff

Искомое значение ключа (256 бит для полнораундового шифрования «Магма»):

ffeeddccbbaa9988 7766554433221100 f0f1f2f3f4f5f6f7 $f8f9fafbfcfdfeff$

Для получения 256 бит ключа необходимо анализировать 8 и больше раундов шифра «Магма» ($n \cdot 32 = 256 \rightarrow n = 8$).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры.
2. *Ищукова Е.А., Богданов К.И., Бабенко Л.К.* Слайдовая атака на криптографический алгоритм Магма и её реализация с использованием технологии параллельного вычисления NVIDIACUDA // Современные наукоемкие технологии. – 2016. – №. 1-1. – С. 25-29.
3. *Babenko L.K., Maro E.A., Anikeev M.V.* Application of algebraic cryptanalysis to Magma and Present block encryption standards // 2017 IEEE 11th International Conference on Application of Information and Communication Technologies (AICT). – IEEE, 2017. – P. 1-7.
4. The Magma Block Cipher [Электронный ресурс] // Github. – URL: https://github.com/istudyatuni/magma/tree/master (дата обращения: 12.07.24)

УДК 004.891.3

И.А. Корх, А.Т. Джамалова

Россия, г. Краснодар, Кубанский Институт информзащиты

АСПЕКТЫ ПРИМЕНЕНИЯ ПОНЯТИЯ «ДОВЕРИЕ» К ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ОРГАНИЗАЦИЙ

Работа продолжает исследования, проводимые в рамках гранта Минобрнауки РФ и посвящена особенностям применения термина «Доверие» в вопросах информационной безопасности при построении систем защиты от атак, проводимых с помощью методов социальной инженерии. Дается сравнительный анализ классической информационной безопасности и социальной инженерии, а также проводится параллель между компонентами технических и киберфизических систем. Приводится результат исследований теоретических аспектов доверия и раскрывается взаимосвязь надежности, устойчивости и психологической составляющей персонала организации, как наиболее уязвимого компонента бизнес-процессов организаций.

Ключевые слова: *социальная инженерия; доверие; информационная безопасность.*

The work continues the research carried out within the framework of a grant from the Ministry of Education and Science of the Russian Federation and is devoted to the peculiarities of using the term "Trust" in information security issues when building protection systems against attacks carried out using social engineering methods. A comparative analysis of classical information security and social engineering is given, and a parallel is drawn between the components of technical and cyberphysical systems. The result of research on the theoretical aspects of trust is presented and the relationship between reliability, stability and the psychological component of the organization's personnel as the most vulnerable component of business processes of organizations is revealed.

Keywords: *social engineering; trust; information security.*

Исследование вопросов доверия и недоверия к компонентам киберфизических является актуальным и междисциплинарным научным направлением, поскольку цифровизация современного общества

вывела данное понятие на новый уровень и дала эмоциональную окраску. Целью данной работы является обоснование подхода и методического приема для изучения доверия и недоверия к информационной безопасности в части персонала организации. Цифровизация общества, как уже рассматривалось в [1], приводит к появлению специфических угроз не только в технических системах, но и системах, главным компонентом которых является человек. Атаки, направленные на персонал организаций, проводятся методами социальной инженерии, которые до недавнего времени игнорировались специалистами по информационной безопасности, а изучались только исследователями гуманитарных и социальных наук. Внесение изменений в паспорта специальностей при подготовке специалистов высшей квалификации в вопросах информационной безопасности дает надежду на совершенствование теоретических основ безопасности с учетом человеческого фактора и развитие научных школ по гуманитарным и социотехническим аспектам информационной безопасности. В работе приводится анализ возможности применения термина «доверие» не только к программным и техническим компонентам систем, но и человеку.

В текущей геополитической обстановке стало абсолютно понятно, что, оказывая влияние на личность, можно до неузнаваемости исказить не только смысл действий, но и настроить по желанию злоумышленника все жизненно важные подсистемы обеспечения безопасности не только личности, но и общества и государства [2]. Необходимость импортозамещения программных и технических решений приводит к изучению надежности внедряемых систем, а также выработке критериев доверия к их информационной безопасности [3].

Доверие к технике, технологии, программному продукту, товару, услуге, человеку, коллеге, персоналу – интуитивно понятный термин, однако, в части информационной безопасности каждого их аспектов применения, при исследовании была выявлена несогласованность, представленная в табл. 1.

Таблица 1

Вид доверия	Кто изучает	Правовые основы
Межличностное доверие и недоверие	Достаточно изучено в работах психологов	Законодательно и методически не закреплено
Доверие и недоверие социальным группам	Достаточно изучено в работах социологов	Законодательно и методически не закреплено
Доверие и недоверие системам	Изучено в части теории надежности	Законодательно и методически не закреплено
Доверие и недоверие технике	Изучается инженерной психологией, психологией труда, эргономики, социальной, экономической психологией	Законодательно и методически не закреплено
Доверие и недоверие технологиям	Влияние ИИ на развитие социотехнических систем	ГОСТ Р 59276-2020 Системы искусственного интеллекта. Способы обеспечения доверия. Общие положения
Доверие к программному продукту	Изучается специалистами ИТ и ИБ	ГОСТ Р 54581-2011 / Информационная технология. Методы и средства обеспечения безопасности. Основы доверия к безопасности ИТ. Часть 1. Обзор и основы

Доверие – отношение человека к другим людям и миру, включающее интерес и уважение к объекту или партнеру, представление о потребностях, которые могут быть удовлетворены в результате взаимодействия с ним, эмоции от предвкушения их удовлетворения и позитивные эмоциональные оценки объекта или партнера, расслаб-

ленность и безусловную готовность проявлять по отношению к нему добрую волю, а также совершать определенные действия, способствующие успешному взаимодействию [4].

Специалисты по человеко-машинным интерфейсам (HMI) занимаются проблемой доверия к технике еще с 1980-х, однако, развитие технологий, имитирующих работу интеллектуальных систем, приводит к появлению «технологических сомнений». Искусственный интеллект с каждым днем охватывает все больше задач, но доверие к выполняемым им операциям не растет, а требует вовлечения человека, что иллюстрирует цикличность развития всех без исключения технологий. Люди строят, углубляют и трансформируют свои отношения с технологиями. Таким образом становится очевидным, что доверие или недоверие к технологиям для каждого конкретного индивида формируется по критериям доверия, что легло в основу разработанных критериев доверия к безопасности ИТ [3]. Кроме того, изучив психологические основы данных понятий, можно сделать вывод о том, что доверие больше основано на эмоциях, а недоверие – на разуме или расчете, что дает основание применять такой подход и к анализу доверия к персоналу. Так, А.Б. Купрейчик выделяет семь категорий, на которые люди распространяют надежды и опасения, когда используют технологии, а в работе предлагается применение семи радикалов для описания подверженности индивида конкретному методу социальной инженерии. Такой подход нашел отражение в работе [5].

Все методы социальной инженерии базируются на конкретных признаках человеческого принятия решения, известных как когнитивные искажения. Такие системные ошибки, которые называются «ошибки в человеческих аппаратных средствах», используются в различных комбинациях для создания различного вида атак. Сравнение «аппаратных и программных ошибок, видов атак, методов тестирования и мер защиты компьютера и человека» приведены в [6].

Наиболее часто для повышения уровня доверия к информационной безопасности со стороны персонала организации прибегают к методу инструктирования [7], который приводит к повышению осведомленности персонала, в том числе.

Решение вопроса взаимосвязи уровня осведомленности и последующего доверия возложено законодательно на владельцев защищаемой информации и соответствующего профильного регулято-

ра. Основные нормативные отраслевые документы, направленные на повышение осведомленности персонала в вопросах ИБ для организаций, приведены в табл. 2.

Таблица 2

Регулятор	Нормативные акты	Статьи
1. Банк России	СТО БР ИББС-1.0 «Обеспечение информационной безопасности организаций банковской системы Российской Федерации. Общие положения» [8]	8.9. Требования к разработке и организации реализации программ по обучению и повышению осведомленности в области информационной безопасности; 8.9.2. Должны быть разработаны планы, программы обучения и повышения осведомленности в области ИБ. По результатам выполнения указанных планов должна осуществляться проверка полученных знаний
2. Медицинские учреждения	Концепция ИБ в сфере здравоохранения (утв. протоколом президиума Правительственной комиссии по цифровому развитию, использованию ИТ для улучшения качества жизни и условий ведения предпринимательской деятельности от 10.03.2022 № 7) [9]	5.17 Регламентация правил и процедур информирования и обучения персонала в области ИБ
	ГОСТ Р ИСО 27799-2015 «Информатизация здоровья. Менеджмент защиты информации в здравоохранении по ИСО/МЭК 27002»	7.5.2.2 Осведомленность, обучение и подготовка в области защиты информации. В дополнение к следованию руководствам, данным в ИСО/МЭК 27002, все организации, занимающиеся обработкой персональной медицинской информации, должны гарантировать предоставление обучения и подготовки в области ЗИ при введении в курс обязанностей, а также то, что регулярные обновления в политике безопасности и методиках организации доводятся до персонала

3. Топливно-энергетический комплекс (ТЭК)	Приказ Министерства энергетики РФ от 6 ноября 2018 г. № 1015 «Об утверждении требований в отношении базовых (обязательных) функций и ИБ объектов электроэнергетики при создании и последующей эксплуатации на территории РФ систем удаленного мониторинга и диагностики энергетического оборудования» (Зарегистрировано в Минюсте РФ 15.02.2019 N№ 53815)	18 В качестве базового набора средств контроля ИБ СУМиД субъект электроэнергетики должен: 1 Утвердить политику ИБ для СУМиД, сформированную в соответствии с инструкцией. 2 Проводить обучение и подготовку персонала по обеспечению ИБ СУМиД; 3 Проводить обучение и подготовку персонала по поддержанию режима информационной безопасности СУМиД
4. Образовательные учреждения	Министерство просвещения Российской Федерации Департамент цифровой трансформации и больших данных ПИСЬМО от 3 марта 2022 года № 04-147 «О мерах по повышению защищенности информационной инфраструктуры системы образования»	Проинформировать администраторов и пользователей информационных систем о недопущении распространения информации о функционировании информационной системы, передаче сторонним лицам своей аутентификационной информации
	Письмо Министерства просвещения РФ от 7 июня 2019 г. № 04-474 «Методические рекомендации по ограничению в образовательных организациях доступа обучающихся к видам информации, распространяемой посредством сети «Интернет», причиняющей вред здоровью и (или) развитию детей, а также не соответствующей задачам образования»	4 Организация просветительской работы с детьми и их родителями по повышению культуры информационной безопасности путем реализации программ и проведения мероприятий в данной области. 5 Направление на повышение квалификации ответственных лиц в образовательной организации по темам «Организация защиты детей от видов информации, распространяемой посредством сети Интернет, причиняющей вред здоровью и (или) развитию детей, а также не соответствующей организациях" и педагогических работников по теме «Безопасное использование сайтов в сети Интернет в образовательном процессе в целях обучения организации»

5. Коммуникационная промышленность	ГОСТ Р 56205-2014 IEC/TS 62443-1-1:2009 Сети коммуникационные промышленные. «Защищенность (кибербезопасность) сети и системы. Терминология, концептуальные положения и модели»	5.7.1 Обучение и повышение осведомленности персонала

Пока человек является пользователем информационных систем, он будет непосредственно влиять и на ее защищенность. Непонимание особенностей взаимодействия элементов и процессов ИС со средствами ЗИ влечет за собой неосознанное повышение угроз информационной безопасности, возникающих вследствие влияния человеческого фактора.

В настоящий момент, данная проблема «неопределённости» пользователей и ответственных по ЗИ не выделяется как отдельная важная проблема, требующая глобальных изменений в подходах к минимизации влияния человеческого фактора, но как показывают статистические данные, полученные в результате анализа наиболее часто фиксируемых инцидентов ИБ за 2023 – 2024 года [23], лучшим решением будет выделить ЧФ, как основную и нерешенную проблему сегодняшнего дня, а доверие – как основной междисциплинарный термин для более внимательного изучения всех критериев, применяемых для анализа компонентов доверия, влияющих на информационную безопасность организаций с учетом человеческого фактора.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Корх И.А., Юмашева Е.В.* Цифровизация и актуальные угрозы информационной безопасности // Наука. Информатизация. Технологии. Образование: Материалы XVI международной научно-практической конференции, Екатеринбург, 27 февраля – 03 2023 года. – Екатеринбург: Российский государственный профессионально-педагогический университет, 2023. – С. 272-281. – EDN ALLSJG.
2. Федеральный закон «О безопасности» от 28.12.2010 N 390-ФЗ (последняя редакция).
3. ГОСТ Р 54581-2011 / Информационная технология. Методы и средства обеспечения безопасности. Основы доверия к безопасности ИТ. Часть 1. Обзор и основы [Электронный ресурс]. – https://docs.cntd.ru/document/1200091394 (дата обращения: 21.07.2024).

4. Ученые записки Ленинградского государственного университета имени А.А. Жданова; № 203. Философский факультет. Серия философских наук. – Вып. 8 «Психология»: Сборник статей / ответственный редактор В.Н. Мясищев.
5. *Корх И.А.* Особенности личности в информационной безопасности // Региональная информатика и информационная безопасность: Сборник трудов XII Санкт-Петербургской межрегиональной конференции, Санкт-Петербург, 27–29 ноября 2021 года. Том Выпуск 10. – СПб.: Региональная общественная организация "Санкт-Петербургское Общество информатики, вычислительной техники, систем связи и управления", 2021. – С. 143-145. – EDN BFNHDN.
6. *Власенко А.В., Корх И.А., Левченко А.А.* Автоматизированная система проведения инструктажей по информационной безопасности организаций // Перспектива-2019: Материалы VIII Всероссийской молодежной школы-семинара по проблемам информационной безопасности, Таганрог, 10–13 октября 2019 года. – Таганрог: ООО «Издательство «Лукоморье», 2019. – С. 134-139. – EDN BPCOMH.
7. *Корх И.А.* Инструктаж, как метод повышения доверия к персоналу // Теория и практика обеспечения информационной безопасности: Сборник научных трудов по материалам всероссийской научно-теоретической конференции, Москва, 03 декабря 2021 года. – М.: Московский технический университет связи и информатики, 2021. – С. 164-171. – EDN HPBIFZ.
8. Стандарт Банка России СТО БР ИББС-1.0-2014 Обеспечение информационной безопасности организаций банковской системы Российской Федерации. Общие положения. (Принят и введен в действие распоряжением Банка России от 17.05.2014 г. № Р-399) // ГАРАНТ.ру: информационной правовой портал [Электронный ресурс]. – URL: https://www.garant.ru/products /ipo/prime/doc/70567254/ (дата обращения: 30.07.2024).
9. Концепция информационной безопасности в сфере здравоохранения (утв. протоколом президиума Правительственной комиссии по цифровому развитию, использованию информационных технологий для улучшения качества жизни и условий ведения предпринимательской деятельности от 10.03.2022 № 7) // Законы, кодексы и нормативно-правовые акты Российской Федерации [Электронный ресурс]. – URL: https://legalacts.ru/doc/kontseptsija-informatsionnoi-bezopasnosti-v-sfere-zdravookhranenija-utv-protokolom-prezidiuma/ (дата обращения: 27.07.2024).

УДК 004.451.25

А.А. Лесников, Е.С. Басан, А.Б. Могильный, М.А. Лыгин, З.А. Быстрая, Д.М. Елькин, М.Г. Шулика

Россия, г. Таганрог, Южный федеральный университет

РАЗРАБОТКА СИСТЕМЫ БЕСПРОВОДНОЙ СВЯЗИ ДЛЯ КИБЕРФИЗИЧЕСКИХ СИСТЕМ

Цель исследования – разработка системы беспроводной связи для киберфизических систем, которая позволяет без дополнительных накладных расходов передавать управляющие команды, координационные данные и изображения между узлами киберфизической системы и оператором. Система связи включает два основных и два резервных канала передачи данных. Тестирование дальности передачи изображения проводилось в условиях прямой видимости между узлами связи и при хороших погодных условиях. Разработанный алгоритм передачи данных позволил добиться устойчивой передачи изображения на расстояние до 2 км в среднем за 7 секунд. Результат исследований – эффективная и надежная система связи, способная обеспечить передачу данных и команд управления в режиме реального времени.

Ключевые слова: *Raspberry Pi 4 Model B; Lora; преобразователь USB в UART; брокер MQTT№ телеметрия; антенна; Jetson Nano.*

The objective of the study is to develop a wireless communication system for cyber-physical systems that allows for the transmission of control commands, coordination data, and images between the nodes of the cyber-physical system and the operator without additional overhead costs. The communication system includes two main and two backup data transmission channels. Testing the image transmission range was carried out under direct visibility between the communication nodes and in good weather conditions. The developed data transmission algorithm made it possible to achieve stable image transmission over a distance of up to 2 km in an average of 7 seconds. The result of the research is an efficient and reliable communication system capable of providing data and control command transmission in real time.

Keywords: *Raspberry Pi 4 Model B; Lora; USB to UART converter; MQTT broker™ telemetry; antenna; Jetson Nano.*

Введение

Система связи представляет собой совокупность технических средств, обеспечивающих передачу данных. Важными компонентами такой системы являются передатчики, приемники, антенны, системы кодирования и декодирования данных, вспомогательное оборудование. Для реализации системы связи был проведен анализ возможных модулей и разработана программно-аппаратная архитектура системы связи, а также проверена дальность передачи изображения в условиях прямой видимости между узлами связи и при хороших погодных условиях.

1. Разработка системы беспроводной связи

1.1. Расчет требований для системы беспроводной связи

На качество и дальность связи влияют несколько факторов:

- мощность передатчика;
- частота передатчика;
- длина и тип антенного кабеля;
- тип передающей антенны и ее размещение;
- препятствия на местности для радиочастот (здания, листья и т.д.);
- тип приемной антенны и ее размещение;
- коэффициент усиления приемной антенны;
- обнаружение несущей приемника;
- чувствительность приемника.

Важным параметром является дальность передачи данных, которая определяется по формуле 1:

$$R = \frac{c}{4\pi F} 10^{\frac{P_{TXdBm}+G_{TXdB}+L_{TXdB}+G_{RXdB}+L_{RXdB}+|V|_{dB}-P_{RXdBm}}{20}}, \quad (1)$$

где R – желаемая дальность связи (м);

$c \approx 3 * 10^8$ – скорость света в вакууме (м/с);

F – частота (Гц);

P_{TXdBm} – мощность наземного антенного блока (дБм);

G_{TXdB} – коэффициент усиления антенны передатчика (дБи);

L_{TXdB} – потери в кабеле от наземного блока до антенны передатчика (дБ);

G_{RXdB} – коэффициент усиления антенны приемника (дБи);

L_{RXdB} – потери в кабеле от наземного блока до антенны передатчика (дБ);

P_{RXdBm} – чувствительность приемника воздушного блока (дБм);

$|V|_{dB}$ – ослабляющий множитель, учитывающий дополнительные потери из-за растительности и т.д.

Формула (1) также показывает, что дальность будет увеличиваться по мере уменьшения частоты передаваемого сигнала F. При этом параметры усиления антенны G_{TXdB} и G_{RXdB} также зависят от частоты. Таким образом, исходя из формулы, можно сделать вывод, что для увеличения дальности передачи необходимо уменьшить частоту и увеличить усиление и мощность передатчика. Кроме того, существенным параметром антенны является dBi. Эта направленность (DN) измеряется в угловых градусах. Чем больше размер направленности антенны, тем уже направленность ее диаграммы [1]. Для увеличения дальности связи используются нестандартные частоты от 868-920 МГц, 433 МГц. Низкие частоты не могут использоваться для передачи цифрового видео и изображений.

Полоса пропускания определяется как диапазон в пределах спектра длин волн, частот или энергии. Это понятие связано с диапазоном радиочастот, занимаемых модулированной несущей.

Автор работы [2] уделяет большое внимание вопросам устойчивой передачи данных, в частности, связности информационного потока или информационных операций. Существует несколько типов электромагнитных волн, которые можно описать синусоидальной функцией, которая характеризуется большой длиной волны. Длина волны λ – это длина периода одного колебания, которая рассчитывается по формуле (2) [3]:

$$\lambda = \frac{c}{f}, \tag{2}$$

где λ – длина волны (м);

$c \approx 3 * 10^8$ – скорость света в вакууме (м/с);

f – частота колебаний ($с^{-1}$).

Формула (2) показывает, что чем выше частота, тем короче длина волны. Низкие частоты с большой длиной волны распространяются лучше на большие расстояния, чем волны с высокой частотой. Это связано с тем, что такие волны могут распространяться вокруг земной поверхности за счет тропосферного распространения. Телевидение и

FM-вещание являются представителями низких частот. Высокие частоты имеют одно главное преимущество – это емкость канала, чем выше диапазон, тем больше каналов можно получить.

Связь между пропускной способностью линии и ее максимально возможной пропускной способностью, независимо от принятого метода физического кодирования, установил Клод Шеннон, которая выражается формулой (3):

$$C = \mathrm{F}\log_2(1 + \frac{P_S}{P_N}), \quad (3)$$

где C – максимальная пропускная способность линии (бит/с);

F – пропускная способность линии (Гц);

P_S – мощность сигнала (Вт);

P_N – мощность шума (дБ).

Полоса пропускания ΔF определяется разностью верхней FB и нижней FH частот в спектре сообщения с учетом его ограничения [4].

Это отношение показывает, что хотя теоретически предела пропускной способности линии с фиксированной полосой пропускания нет, на практике он есть. Действительно, можно увеличить пропускную способность линии, увеличив мощность передатчика или уменьшив мощность шума (помех) на линии связи. Кроме того, влияние мощности полезного сигнала и шума на пропускную способность ограничено логарифмической зависимостью, которая растет не так быстро, как прямо пропорциональная зависимость. Например, при довольно типичном начальном отношении мощности сигнала к мощности шума в 100 раз, увеличение мощности передатчика в 2 раза даст лишь 15% увеличение пропускной способности линии [5].

Индикатор уровня принятого сигнала (RSSI) – это полная мощность сигнала, принимаемого приемником. Он измеряется приемником в логарифмическом масштабе в dBm (dBm – децибел относительно 1 милливатта) и определяется по формуле (4) [6]:

$$RSSI = P_0 - 10n\lg\left(\frac{d}{d_0}\right), \quad (4)$$

где $RSSI$ – полная мощность сигнала, принимаемого приемником (dBm);

P_0 – мощность сигнала прибора, измеренная на единичном расстоянии $d0$ от прибора (dBm);

n – коэффициент потерь мощности сигнала при распространении в среде, безразмерная величина (для воздуха n=2; увеличивается при наличии препятствий);

d – расстояние от прибора до передатчика (м);

d_0 – расстояние от прибора до точки, в которой измерялась мощность сигнала (м).

SNR указывается либо в dBc (дБ на несущую), когда в качестве эталона используется абсолютная мощность основной частоты, либо в dBFS (дБ полной шкалы), когда мощность основной частоты экстраполируется на полный диапазон преобразователя (5):

$$SNR = 10\lg\left(\frac{P_S}{P_N}\right). \quad (5)$$

Интенсивность передачи данных определяется по следующему принципу: N раз опрашивается канал связи и если в этот момент на канале обнаружена передача данных, то канал считается активным (A) (6):

$$IS = \sum_N A. \quad (6)$$

Таким образом, при расчете параметров системы связи необходимо учитывать следующие параметры, зависимость между которыми можно выразить условной функцией H, через расчет параметров системы связи CS (7):

$$CS = H(G_{db}, P_{db}, f, C). \quad (7)$$

Другим важным параметром является угол направленности антенны для увеличения дальности приема. Хотя этот параметр отсутствует в уравнении дальности, его необходимо учитывать, поскольку наземные станции обычно используют направленные антенны. При максимальной дальности связи в системах передачи видеосигнала также достигнуты успехи в использовании кабелей с минимальным линейным затуханием и максимальном ограничении их длины. Для монтажа системы и передачи видеоизображения следует учитывать, что и наземная станция (НС), и приемопередающие модули должны быть установлены как можно ближе к антеннам, чтобы минимизировать передачу высокочастотного сигнала по кабелям. Одним из возможных антенных устройств может быть дипольное антенное устройство. Дипольное антенное устройство имеет следующие характеристики [7]:

- дальность действия – от 0,3 км до 0,4 км;
- площадь покрытия – от 0,282 км^2 до 0,502 км^2;
- угол действия – 360 градусов (плоскость);

- сопротивление – 50 Ом;
- коэффициент усиления – 3 dBi.

Другим возможным антенным устройством может быть антенное устройство «Биквадрата Харченко» [8]. Антенное устройство «Биквадрата Харченко» обеспечивает следующие характеристики:

- дальность действия – до 2 км;
- площадь покрытия – 4,88 км^2;
- угол действия – от 140 градусов;
- сопротивление – 50 Ом;
- коэффициент усиления – 7 dBi.

1.2. Архитектура системы беспроводной связи

Связь для передачи команд и телеметрии организована с помощью модулей LoRa на стороне оператора и модема LoRa на другой стороне [9]. С помощью модулей достигается гарантированная дальность связи 1-2 км без использования дополнительных усилителей или ретрансляторов сигнала. Связь для передачи изображения организована с помощью радиомодулей LoRa WAN, так как они обладают наибольшей скоростью передачи данных (128 кбит/с) и имеют необходимую дальность связи (до 3 км). Поскольку задача передачи изображения отличается по аппаратным требованиям от задачи передачи телеметрии, была разработана следующая архитектура системы связи, которая представлена на рис. 1.

Данная архитектура обеспечивает бесперебойную и надежную связь, так как передача изображений и управление разнесены по разным модулям связи. Кроме того, обеспечивается частотное разделение. Так, передача команд и телеметрии [10] осуществляется с помощью радиомодулей, работающих в диапазоне частот от 850 МГц до 930 МГц, а передача изображений и подтверждений их получения осуществляется с помощью радиомодулей, работающих в диапазоне частот от 425 до 450 МГц. Дальность передачи составляет 3 км в условиях прямой видимости и отсутствия существенных помех и естественных наложений каналов связи в диапазонах частот модулей. При этом гарантия передачи команд и изображений также обеспечивается алгоритмами обмена данными и подтверждения приема сообщений. С учетом разработанной модели системы связи, выбранных модулей радиосвязи [11] была реализована итоговая архитектура связи, учитывающая как программную реализацию модулей передачи и приема: команд, телеметрии и изобра-

жений, так и аппаратную реализацию в виде полнодуплексных радиостанций [12]. Общая архитектура системы связи с учетом программных модулей и аппаратных компонентов представлена на рис. 2.

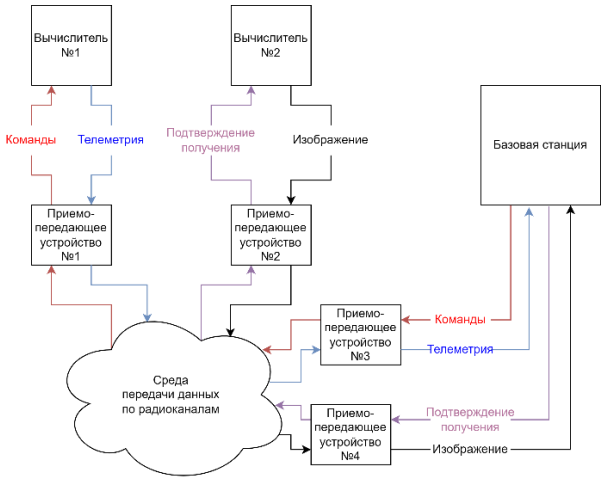


Рис. 1. Архитектура системы беспроводной связи

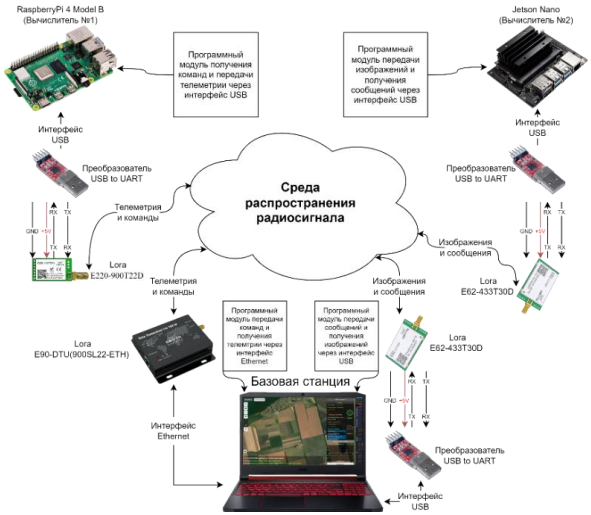


Рис. 2. Программная и аппаратная архитектура системы беспроводной связи

2. Алгоритм работы модуля связи

Радиомодемы подключаются к компьютерам и базовой станции с помощью преобразователей USB-UART [13]. Радиомодуль подключается к базовой станции через интерфейс Ethernet по протоколу TCP или UDP, но требует внешнего питания напряжением от 8 до 28 вольт. Для передачи данных на указанных радиомодулях были разработаны программные модули передачи и приема на языке программирования Python [14]. Общий алгоритм работы модуля показан на рис. 3.

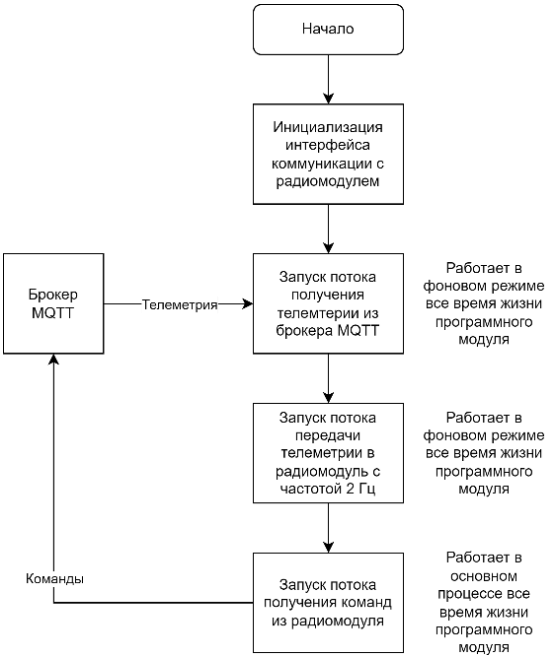


Рис. 3. Общий алгоритм работы программного модуля приема команд и передачи телеметрии по радиоканалу

При инициализации подключения к радиостанции создается TCP-сервер. Далее радиомодуль LoRa подключается к созданному TCP-серверу. Алгоритмы передачи, формирования, кодирования и декодирования сообщений аналогичны алгоритмам приема сообщений, команд и полетных заданий, описанным выше. Поток команд и полет-

ных заданий от брокера MQTT извлекает команды и полетные задания, а затем передает их в формате, определяемом системой связи между двумя модулями по радиоканалу. Формат кодирования и декодирования сообщений с командами и полетными заданиями одинаков для передающего и принимающего модулей. Поток приема телеметрии принимает сообщения по радиоканалу, декодирует данные и восстанавливает исходный формат, а также публикует восстановленные телеметрические сообщения в операторском модуле на брокере MQTT [15].

При вызове функции передачи изображения, входными данными которой является массив байт изображения, массив фрагментируется на блоки по 220 байт, которые в дальнейшем будем называть пакетами изображения [16]. Далее определяется количество таких пакетов, чтобы обеспечить передачу каждой части изображения через радиомодуль в модуль оператора. Алгоритм передачи пакетов изображения заключается в синхронной передаче каждого из пакетов изображения и предварительной упаковке этих данных в сообщение с пачкой изображений. При этом проверяется наличие сообщения с ответом на частоте 100 Гц, но если в течение 2,5 секунд не удается отправить еще один пакет изображения, то алгоритм прерывается и все данные связи с радиомодулем очищаются, после чего передача передачи изображения повторяется, начиная с шага вызова функции передачи [17].

Таким образом, разработанная система связи, передающая команды, сообщения, полетные задания и изображения, обеспечивает необходимую дальность связи до трех километров, а также выполняет весь функционал. Для проверки дальности связи проводилась передача изображений, так как она является наиболее ресурсоемкой и требовательной к качеству сигнала.

3. Экспериментальное исследование

На стороне оператора и на другой стороне используется вертикально поляризованная всенаправленная антенна. Эта антенна предназначена для передачи сигналов в диапазоне частот 433 МГц.

Для тестирования дальности обе радиостанции были настроены следующим образом: скорость обмена данными UART – 9600 бит/с, скорость передачи данных по воздуху – 64 кбит/с, мощность – 30 дБм, диапазон частот от 433,0 МГц до 434,0 МГц.

Тестирование дальности передачи изображений проводилось в условиях прямой видимости между узлами связи, а также в хороших погодных условиях [18]. Двадцать изображений передавались с одного радиоузла на другой, в результате чего определялась средняя скорость передачи изображений, а также характеристики качества сигнала. Качество сигнала определялось количеством повторных попыток одного или нескольких пакетов изображений. Чем меньше попыток, тем стабильнее соединение. Измерения дальности связи проводились на расстояниях: 100 метров, 200 метров, 400 метров, 600 метров, 1000 метров, 1400 метров, 2000 метров, 2200 метров.

На рис. 4 показаны контрольные точки измерений дальности связи, в которых проверялась передача изображения, а также место расположения тестового образца модуля связи, принимающего сигнал (оператора).

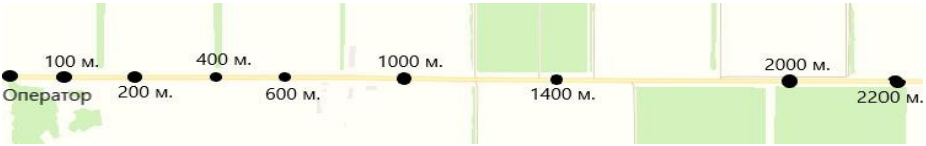


Рис. 4. Схема контрольных точек для проверки дальности передачи изображения

Антенна модуля оператора была установлена на высоте 2,5 метра. Передача изображений осуществлялась с помощью программы проверки дальности, которая регистрировала испытания, а также вычисляла значения, характеризующие качество сигнала. Вывод программы при проверке дальности связи на расстоянии 1,4 км от модуля оператора.

На основании данного теста следует, что на расстоянии 1,4 км качество сигнала хорошее, так как 20 из 20 изображений были доставлены успешно. Время передачи 20 изображений составило 96,54 секунды, что составляет примерно 4,8 секунды на изображение. Средняя скорость передачи изображений составила 0,65 КБ/с. Анализ логов данного теста показал, что количество повторных попыток передачи пакетов изображений составило 4 попытки. Общее количество пакетов всех 20 изображений составило 265. Таким образом, процент потери пакетов составляет 1,5%. Таким образом, на

основании результатов всех проведенных тестов была составлена табл. 1, описывающая показатели качества сигнала в зависимости от расстояния передачи.

Таблица 1

Результаты испытаний дальности связи

Тест, №	Диапазон, м	Всего пакетов, пак	Время передачи, с	Скорость передачи изображения, Кбит/с	Количество повторных попыток отправки пакетов, пак	Потеряно пакетов, %	Передано изображений, %
1	100	260	94,7	0,68	0	0	100
2	200	270	95,3	0,67	0	0	100
3	400	257	94,2	0,64	0	0	100
4	600	280	99,3	0,65	1	0,36	100
5	1000	275	95,5	0,66	3	1,1	100
6	1400	265	96,54	0,65	4	1,5	100
7	2000	257	94,4	0,64	6	1,6	97
8	2200	281	194,1	0,33	12	2,8	90

Вывод

Таким образом, была разработана и протестирована система. Использован общий алгоритм работы модуля передачи радиоизображений. Передача изображений осуществлялась с помощью программы-дальномер, которая выполняла тестовую регистрацию, а также расчет значений. Рассмотрев результаты теста дальности связи, можно сделать вывод, что процент переданных изображений начал падать только при настройке 2200 метров.

Благодарности. *Исследование выполнено при поддержке НИР № ВнГр/24-01-КТ «Разработка демонстрационной модели БПЛА с повышенной отказоустойчивостью» пункты 2,3, а также при поддержке Совета по грантам Президента Российской Федерации за счет средств стипендии Президента Российской Федерации для мо-*

лодых ученых и аспирантов (Конкурс СП-2022) № СП-858.2022.5 по теме «Технология обеспечения кибербезопасности автоматизированных систем от активных информационных атак на основе принципа отражения», пункт 1.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Lockheed-Martin K.C.* Boeing F-22 Raptor. Assessing the F-22A Raptor / AFAIAA, SMIEEE, PEng. – 2007. – URL: http://www.ausairpower.net/APA-Raptor.html (дата обращения: 12.07.2024).
2. *Ivanov V.K.* Physics. Electromagnetic waves: textbook. Manual. – St. Petersburg: POLYTECH-PRESS, 2023. – 208 p.
3. Appendix No. 3 to the decision of the State Commission on Radio Frequencies dated November 29, 2021 No. 21-60-01. – URL: https://digital.gov.ru/uploaded/files/prilozhenie-3-k-resh-21-60-01.pdf (дата обращения: 12.07.2024).
4. *Катунин Г.П. Мамчев Г.В., Попантонопуло В.Н., Шувалов В.П.* Телекоммуникационные системы и сети. В 3 т. Т. 2. Радиосвязь, радиовещание, телевидение [Текст]. – Горячая линия – Телеком, 2018. – 564 с.
5. *Бухтияров Д.А., Горбачев А.П.* Исследование дипольной антенны с концевым возбуждением, питаемой прямоугольным волноводом // Известия высших учебных заведений. Радиофизика. – 2017. – Т. 60, № 1. – С. 32-40. – EDN YINZTF.
6. *Харченко К.П.* Антенна диапазона ДЦВ // В помощь радиолюбителю. – 1986. – Вып. 94. – С. 68-79.
7. *Bishop M.* Introduction to Computer Security. – 1st ed. – Addison-Wesley, Ed. Boston, USA: Pearson Education, 2004.
8. *Young C.* Metrics and Methods for Security Risk Management: Syngress Media, 2010.
9. General properties of Lora. – URL: https://help.mikrotik.com/docs/display/ROS/Lora (дата обращения: 12.07.2024).
10. NVIDIA Jetson XAVIER NX Developer Kit. 2021. – URL: https://developer.nvidia.com/embedded/jetson-XAVIER-NX-developer-kit/ (дата обращения: 12.07.2024).
11. *Nonami K.* Prospect and recent research & development for civil use autonomous unmanned aircraft as UAV and MAV // J.Syst.Des.Dyn. – 2007. – 1. – P. 120-128.
12. *Peniak P., Holečko P., Bubeníková E. and Kanáliková A*. LoRaWAN Sensors Integration for Manufacturing Applications via Edge Device Model with OPC UA» [Text] // 2023 International Conference on Applied Electronics (AE), Pilsen, Czech Republic, 2023. – P. 1-6, – DOI: 10.1109/AE58099.2023.10274274.

13. RaspberryPi documentation. – URL: https://raspberrypi.ru/doc/ (дата обращения: 12.07.2024).
14. *Anand M., Asok Kumar A., Athul B. Francis, Karthika Mohan, Midhun R., Mohamed Samshad.* Short Range Telemetry Communication for Autonomous Drone Navigation // 2020 IEEE Recent Advances in Intelligent Computional Systems, India, 2020. – P. 131-135.
15. Broker Mosqiutto MQTT. – URL: https://mosquitto.org/ (дата обращения: 12.07.2024).
16. *Cetin O., Zegli I., Yilmaz G.* Establishing obstacle and collision free communication relay for UAVs with artifical potential fields // J. Intell. Robot. Syst. – 2013. – 69. – P. 361-372.
17. *Brown T.X., Doshi S., Jadhav S., Himmelstein J.* Test Bed for a Wireless Network on Small UAVs // In Proceedings of the AIAA 3 rd Unmanned Unlimited Technical Conference, Chicago, IL, USA, 20-23 September 2004. – P. 20-23.
18. *Asadpour M., Giustiniano D., Hummel K.A., Heimlicher S., Egli S.* Now or Later ? – Delaying Data Transfer in Time-Critical Aerial Communication // In Proceedings of the 2013 ACM International Conference on Emerging Networking Experiments and Technologies, New York,USA,14-19 December 2013. – P. 127-132.

УДК 004.056

В.О. Малявина, Е.А. Маро

Россия, г. Таганрог, Южный федеральный университет

МОДЕЛИРОВАНИЕ УТЕЧЕК ПО ПОБОЧНЫМ КАНАЛАМ ДЛЯ КРИПТОГРАФИЧЕСКОГО АЛГОРИТМА «МАГМА» НА ОСНОВЕ ЭМУЛЯТОРА ELMO

В исследовательской работе с помощью инструмента ELMO получены трассы энергопотребления для алгоритма шифрования «Магма» и выявлены инструкции, содержащие статистические утечки по энергопотреблению. Для моделирования трасс энергопотребления в ELMO реализован на языке C алгоритм шифрования ГОСТ Р 34.12—2015 (n=64 «Магма»). Выполнено моделирование утечек по энергопотреблению для различного числа раундов шифрования «Магма» на основе t-теста. Выявленные инструкции являются оптимальными для последующего проведения дифференциальных или корреляционных атак по энергопотреблению на исследуемый алгоритм шифрования.

Ключевые слова: *моделирование утечек по энергопотреблению; эмулятор ELMO; симметричный блочный алгоритм шифрования; ГОСТ Р 34.12-2015; шифр «Магма».*

In this research, the ELMO tool was used to obtain power consumption traces for the Magma encryption algorithm and identify instructions containing statistical power consumption leaks. To model the power consumption traces, the GOST R 34.12—2015 encryption algorithm (n=64 Magma) was implemented in C in ELMO. Power consumption leaks were modeled for different numbers of Magma encryption rounds based on the t-test. The identified instructions are optimal for subsequent differential or correlation attacks on power consumption on the observed encryption algorithm.

Keywords: *power consumption leak modeling; ELMO emulator; symmetric block encryption algorithm; GOST R 34.12-2015; Magma cipher.*

Введение

Классический криптоанализ симметричных шифров рассматривает криптосистему как математический алгоритм, преобразующий некоторый входной текст (или наборы входных текстов) в вы-

ходной текст (соответствующий набор выходных текстов) на основе исследователь имеет полное описание преобразований, происходящих внутри криптосистемы, владеет зашифрованными текстами, может обладать соответствующими открытыми текстами (или их частями), но не обладает информацией об используемом секретном ключе. Классические методы криптоанализа опираются на использование недостатков математической конструкции шифра для вычисления ключа шифрования по известным данным, вычислительно быстрее полного перебора множества возможных значений ключей.

На практике криптографический алгоритм не ограничивается только математическим описанием алгоритма шифрования, так как не может существовать без физической реализации в виде конкретного программного или программно-аппаратного средства. Криптографический алгоритм разработан в определенной программной среде, реализуется на определенном оборудовании (типе процессора), что отражается на специфике работы криптосредства и может быть использовано исследователем при криптоанализе.

Атаки по побочным каналам

Атаки по побочным каналам представляют собой класс атак, направленный на использование уязвимости (недостатка) в практической реализации криптосистемы. Учитывая важность анализа безопасности различных реализаций криптографических систем, следует отдельно рассматривать стойкость средства защиты информации к атакам по побочным каналам [1, 2]. Классификация атак по побочным каналам [3] приведена на рис. 1.

Первоначальным этапом оценки стойкости реализаций криптографических средств защиты к атакам по побочным каналам является выявление утечки, присущей работе криптосистемы. Одним из универсальных каналов утечки для криптографических систем служит канал энергопотребления. Атака по энергопотреблению – пассивная атака, направленная на выявление зависимости между энергопотреблением шифратора (процессора) и преобразуемыми данными с целью получения секретного ключа или защищаемой информации. При проведении атаки по энергопотреблению исследователь должен иметь возможность выполнять измерения энергопотребления с высокой точностью для получения информации о выполняемых на устройстве операциях и их параметрах. Типичная схема стенда для

проведения атаки по энергопотреблению показана на рис. 2. Выделяют следующие разновидности атак, в которых используется информация об энергопотреблении: простой анализ энергопотребления, дифференциальный анализ энергопотребления, корреляционный анализа энергопотребления и анализ на основе шаблонов.

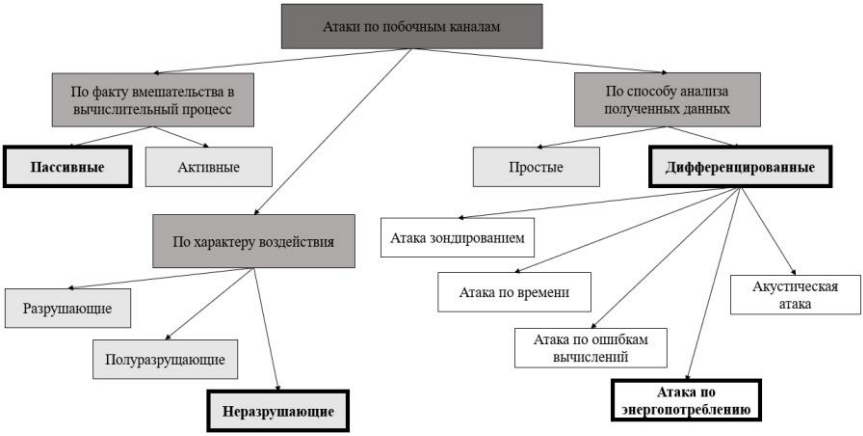


Рис. 1. Классификация атак по побочным каналам

Рис. 2. Структура стенда для проведения атаки по энергопотреблению

Стойкость реализации к утечкам по энергопотреблению рассматривается как важная составляющая обеспечения заданного уровня безопасности и доверия к средству защиты информации в целом.

В данном исследовании проведено моделирование трасс энергопотребления криптографических средств защиты информации, в основе которых используется реализация алгоритма ГОСТ Р 34.15-2015 [4] (n=64 «Магма»), и выполнен поиск наличия каналов утечки, путем выявления наборов инструкций, для которых имеются статистические зависимости энергопотребления устройства от значения обрабатываемых данных (по результатам t-теста).

Моделирование энергопотребления с помощью инструмента ELMO

Любое моделирование энергопотребления (или другого побочного канала) состоит из двух составных частей: эмуляция процесса, который выполняется внутри устройства, и моделирование наблюдаемого извне поведения устройства (для сопоставления эмулируемых процессов с прогнозируемым потреблением). Моделирование можно в общих чертах разделить на категории в зависимости от архитектурного уровня, на котором они пытаются охарактеризовать мощность:

1. Моделирование транзисторного уровня. При наличии достаточной информации о технологии, по которой будет построен чип, схему можно сопоставить с сетью транзисторов, потребляемая мощность которых моделируется с помощью известных дифференциальных уравнений.

2. Моделирование уровня шлюза. Данный вид также основан на списках соединений (с обратной аннотацией). Для моделирования количество переходов в каждом шлюзе подсчитывается и взвешивается в соответствии с информацией в списке соединений. Тогда сумма по всем взвешенным переходам является приближением мгновенной мощности схемы.

3. Моделирование поведенческого уровня. На этом уровне нет информации о размещении элементов схемы и маршрутизации сигналов между ними. Доступ имеется только к поведенческому описанию компонентов – например, в форме машинного кода или микрокода низкого уровня, кода инструкции/ассемблера или кода более

высокого уровня (например, C). Разработка точных моделей на этом уровне требует доступа к реальным устройствам (и лабораторной установке), с помощью которых оценивается средняя мощность различных (последовательностей) инструкций. Они хорошо подходят для небольших устройств (и, следовательно, низкой сложности), в которых инструкции по сборке сопоставляются непосредственно с машинными инструкциями без дальнейшего декодирования в микроинструкции.

ELMO [5] – это инструмент моделирования, совмещающий несколько категорий из описанных выше. Инструмент ELMO содержит в себе эмулятор конкретной архитектуры (Arm Cortex-M0) с эмпирически оцененными моделями энергопотребления, зависящими от входных данных.

Программное средство ELMO основано на эмуляторе инструкций для Thumb под названием Thumbulator. Thumbulator предназначен для воспроизведения работы процессоров семейства ARM Cortex M0 при выполнении симметричных блочных шифров. Thumbulator принимает на вход двоичную программу на ассемблере Thumb и транслирует ее в машинные инструкции, что позволяет воспроизвести поток данных ядра микропроцессора с достаточной точностью [6].

Для моделирования энергопотребления криптосистем, основанных на ARM Cortex M0, в ELMO интегрированы широко используемые в реализациях симметричной криптографии модели инструкции Thumb. Эти инструкции можно распределить на 5 разных групп:

1) инструкции загрузки (ldr, ldrb, ldrh);
2) инструкции ALU (adds, adds #imm, ands, eors, movs, movs #imm, orrs, subs, subs #imm, cmp, cmp #imm);
3) инструкции сохранения (str, strb, strh);
4) инструкции сдвига (lsls, lsrs, rors);
5) инструкция умножения (muls).

Инструкции с утечкой получены в результате выполнения статистического поиска на основе t-теста в ELMO. Значение параметра t вычисляется по формуле (1):

$$t = \frac{mean_{fix} - mean_{rand}}{\sqrt{\frac{var_{fix}}{N_{fix}} + \frac{var_{rand}}{N_{rand}}}}, \quad (1)$$

где *fix* – группа фиксированных значений;

rand – группа случайных значений;

mean – среднее значение всех трасс в группе;

var – стандартное отклонение выборки всех трасс в группе;

N – размер группы.

Пороговым значением, по которому делается вывод о наличии утечки, зафиксировано значение t = |4.5|, в соответствии с рекомендациями стандарта CSA ISO/IEC 17825-2018 "Information technology - Security techniques - Testing methods for the mitigation of non-invasive attack classes against cryptographic modules" [7] по использованию Test Vector Leakage Assessment (TVLA) [8, 9].

Моделирование энергопотребления шифра «Магма»

В алгоритме «Магма» для шифрования используется блок размером 64 бита, длина ключа составляет 256 бит. Раундовые ключи получаются из исходного путем его деления на восемь 32-битных подключей (Ki). После получения подключей идет непосредственно процесс шифрования: блок входных данных разделяется на две равные по длине части – правую (R) и левую (L) (по 32 бита каждая), над которыми выполняется тридцать две итерации раундового преобразования с использованием раундовых ключей. На рис. 3 представлена схема раундового преобразования алгоритма «Магма» при шифровании.

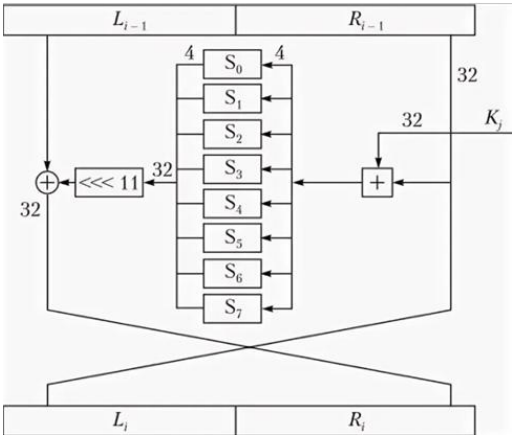


Рис. 3. Схема раундового преобразования алгоритма «Магма»

Проведена оценка общего количества инструкций и количества инструкций, содержащих статистические утечки по энергопотреблению, для различного числа раундов. Результаты моделирования представлены в табл. 1.

Таблица 1

Результаты моделирования утечек по побочным каналам для различного количества раундов шифра «Магма»

Кол-во раундов	Общее кол-во инструкций	Инструкции с утечками по энергопотреблению	Процентное соотношение
32	15400	4450	29%
16	8820	2203	25%
2	2992	462	15%
1	2599	277	11%

На рис. 4 представлена одна трасса энергопотребления полнораундового шифра «Магма». Из данного графика видно, как сначала выполняются служебные инструкции инициализации параметров шифрования (независимые от количества раундов), после чего запускаются группы инструкций раундового преобразования (соответствующие 32 пика).

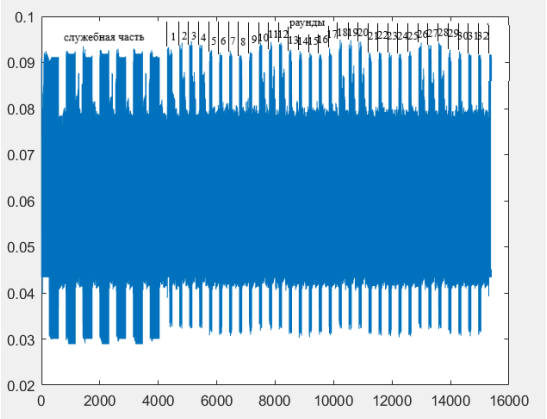


Рис. 4. Сгенерированная с помощью инструмента ELMO трасса энергопотребления при выполнении тридцати двух раундов алгоритма «Магма»

На рис. 5 представлены результаты теста FixedvsRandom для одного раунда шифра «Магма». Наиболее уязвимые для атак посторонним каналам инструкции занесены в табл. 2, курсивом выделены два наибольших значения по результатам статистического теста.

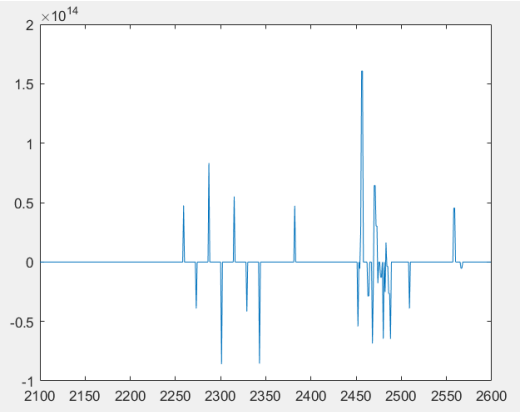


Рис. 5. Результаты применения статистического теста FixedvsRandom в эмуляторе ELMO для одного раунда шифра «Магмы»

Таблица 2

Инструкции, содержащие статистическую зависимость (утечку), для одного раунда шифра «Магма»

Номер инструкции	Адрес	Машинный код	Ассемблерный код инструкции
2287	0x0800014A	0x090B	lsrs r3,r1,#0x4
2301	0x0800014A	0x090B	lsrs r3,r1,#0x4
2343	0x0800014A	0x090B	lsrs r3,r1,#0x4
2456	*0x08000176*	*0x18E3*	*adds r3,r4,r3*
2457	*0x08000178*	*0x5C5B*	*ldrb r3,[r3,r1]*
2468	0x08000172	0x011B	lsls r3,r3,#0x4
2470	0x08000176	0x18E3	adds r3,r4,r3
2471	0x08000178	0x5C5B	ldrb r3,[r3,r1]
2480	0x0800016E	0x090B	lsrs r3,r1,#0x4
2488	0x0800017A	0x5483	strb r3,[r0,r2]

Заключение

В рамках исследования проведено моделирование трасс энергопотребления для различного количества раундов алгоритмов шифрования «Магма». Выделены сигнатуры трасс энергопотребления, соответствующие отдельным раундам и преобразованиям шифрования исследуемого алгоритма. Выполнены наборы статистических тестов (t-тестов), по которым определены инструкции, содержащие утечки для одного раунда шифров «Магма». Сформированные файлы трасс энергопотребления, ассемблерного кода и FxedvsRandom тестов предоставлены в общий доступ для возможности ознакомления и последующего использования результатов моделирования.

Полнораундовая версия алгоритма шифрования «Магма» содержит 15400 инструкций, из них 4450 инструкций содержит потенциальную утечку по энергопотреблению. Для одного раунда шифра «Магма» инструкции с номерами 2456 (adds r3,r4,r3) и 2457 (ldrb r3,[r3,r1]) имеют максимальное значение по статистическому тесту FixedvsRandom. Выявленные инструкции и следовательно точки на трассах энергопотребления являются оптимальными для проведения анализа стойкости исследуемой реализации алгоритма шифрования к атаке по побочному каналу энергопотребления.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Hou X., Breier J.* Side-Channel Analysis Attacks and Countermeasures // In: Cryptography and Embedded Systems Security. – Springer, Cham. – https://doi.org/10.1007/978-3-031-62205-2_4.
2. *Piessens F. and van Oorschot P.C.* Side-Channel Attacks: A Short Tour // in IEEE Security & Privacy. – March-April 2024. – Vol. 22, No. 2. – P. 75-80. – DOI: 10.1109/MSEC.2024.3352848.
3. *Krasovsky A.V. and Maro E.A.* Actual and historical state of side channel attacks theory // In Proceedings of the 12th International Conference on Security of Information and Networks (SIN '19). Association for Computing Machinery, New York, NY, USA, Article 13. – P. 1–7. – https://doi.org/10.1145/3357613.3357627.
4. ГОСТ Р 34.12-2015 «Информационная технология. Криптографическая защита информации. Блочные шифры» [Электронный ресурс]. – URL: https://tc26.ru/standard/gost/GOST_R_3412-2015.pdf
5. Statistical leakage simulator for the ARM M0 family ELMO [Электронный ресурс]. – URL: https://github.com/scaresearch/ELMO.

6. *Welch D.* Thumbulator [Электронный ресурс]. – URL: https://github.com/dwelch67/thumbulator.git.
7. CSA ISO/IEC 17825-2018 Information technology - Security techniques - Testing methods for the mitigation of non-invasive attack classes against cryptographic modules.
8. *Goodwill G., Jun B., Jaffe J. and Rohatgi P*. A testing methodology for side-channel resistance validation, NIST Non-Invasive At-tack Testing Workshop, 2011.
9. *Cooper J., DeMulder E., Goodwill G., Jaffe J., Kenworthy G. and Rohatgi P.* Test vector leakage assessment (tvla) methodology in practice // International Cryptographic Module Conference. – 2013.

УДК 004.051

И.В. Машкина, А.М. Уразаева

Россия, г. Уфа, Уфимский университет науки и технологий

РАЗРАБОТКА МОДУЛЯ БАЗЫ ЗНАНИЙ СЦЕНАРИЕВ УГРОЗ ДЛЯ СИСТЕМЫ РЕАГИРОВАНИЯ НА ИНЦИДЕНТЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ (IRP)

Целью работы является исследование возможности повышения эффективности реагирования на инциденты информационной безопасности (ИБ). Приведена архитектура системы реагирования на инциденты, а также методика разработки базы знаний сценариев многокомпонентных атак с учетом тактик, техник, уязвимостей и угроз безопасности информации (УБИ), приведенных в нормативных документах, на основе EPC-моделирования формализована оценка вероятности реализации сценария.

Ключевые слова: *система реагирования на инциденты; тактики; техники; уязвимости; УБИ; база знаний сценариев угроз; база знаний сценариев реагирования; EPC-диаграмма.*

The aim of the article is to study the possibility of improving efficiency of information security incident response. Incident Response System architecture, along with the methodology for construction of multicomponent threat scenarios knowledge base have been developed. Scenarios are designed on the basis of the regulatory documents taking into account tactics, technics, vulnerabilities and information security threats. Probability estimation of scenario realization is formalized based on EPC-modeling.

Keywords: *incident response system; tactics; technics; vulnerabilities; information security threats; threat scenarios knowledge base; EPC -diagram.*

Приказы ФСТЭК России №31 и №239 утверждают требования к обеспечению защиты информации в АСУ ТП на критически важных и потенциально опасных объектах: атомной энергетики, добычи и транспортировки нефти и газа, оборотно-промышленного комплекса и транспорта [1, 2]. Специалисты в области ИБ отмечают, что в 2024 году проблемы защиты АСУ ТП остаются крайне актуальными и сложными [3–5]. Среди ключевых проблем, таких как: интеграция АСУ ТП с кор-

поративными ИТ-сетями, увеличение числа целенаправленных атак на системы управления технологическими процессами, необходимость обеспечения совместимости средств защиты информации с импортозамещающими отечественными SCADA-системами и ПЛК, – особо отмечаются нереализованные на многих объектах процессы мониторинга и реагирования на инциденты.

Отсутствие системы мониторинга, анализа угроз и реагирования на инциденты может привести к нарушению киберустойчивости промышленной системы и, следовательно, к нарушению непрерывности технологического процесса, увеличению времени на восстановление после атаки.

В последние годы особую актуальность приобрела тематика автоматизации реагирования на угрозы ИБ. Некоторые SIEM обладают встроенной IRP-системой, способной быстро локализовать инцидент и уменьшить или исключить разрушительность последствий.

Несколько самостоятельных IRP решений российского производителя для выполнения базовых задач находятся в промышленной эксплуатации: Jet Signal компании «Инфосистемы Джет», R-Vision компании «Р-Вижн», Security Vision компании «Интеллектуальная безопасность» [6–8]. На российском рынке представлены также продукты Израильской компании CyberBit SOC 3D, а также разработка компании IBM IBM Resilient IRP [9, 10].

IRP (Incident Response Platform) – это система автоматизации реагирования на инциденты кибербезопасности, которая выполняет функции по сбору дополнительной информации, сдерживанию, устранению угрозы либо восстановлению системы после атаки, а также по структурированию данных о расследовании инцидента [11].

На рис. 1 приведена предлагаемая архитектура построения IRP системы. Основными модулями являются база знаний сценариев реагирования и база знаний сценариев угроз. Причем база сценариев реагирования может быть разработана на основе полного перечня всех возможных типов инцидентов ИБ, т.е. на основе модели угроз конкретному объекту защиты, которая, как известно [12], включает в себя список актуальных угроз. Таким образом, эффективные сценарии реагирования на киберинциденты могут быть разработаны только с учетом сценариев угроз. На основе модели сценария угрозы формируется адекватный сценарий реагирования, уникальный для

каждой последовательности событий и задействованных объектов. Сценарий реагирования представляет собой совокупность правил и выполняемых действий, специфичных для индикаторов – признаков этапов реализуемого сценария угрозы.

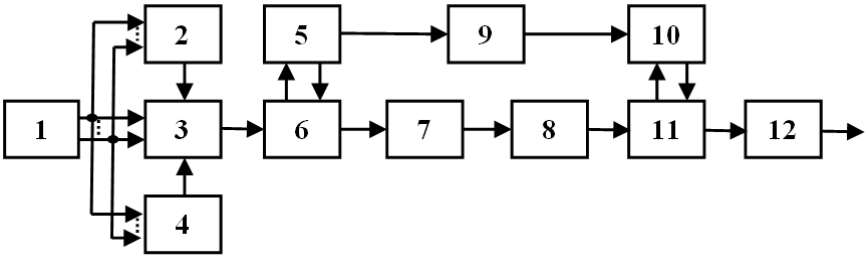


Рис. 1. Архитектура системы реагирования на инциденты

1 – источники данных о событиях безопасности (с SIEM системы);

2 – модуль определения задействованных объектов инфраструктуры;

3 – модуль анализа данных (техник, индикаторов и др.);

4 – модуль контроля привилегий;

5 – база знаний сценариев угроз целевым объектам инфраструктуры;

6 – модуль определения сценария угрозы в реальном времени;

7 – модуль численной оценки риска реализации угрозы;

8 – модуль определения статуса инцидента;

9 – модуль разработки плана реагирования;

10 – база знаний сценариев реагирования, адаптированных под конкретные сценарии угроз;

11 – модуль принятия решений;

12 – модуль формирования командной информации (Active Response) на агенты реагирования (запуск скриптов, воздействие на средства защиты и др.).

Рассмотрим, как может быть решена задача разработки сценариев угроз для создания базы знаний угроз и сценариев реагирования на их основе. В работах [13, 14] было предложено для моделирования угроз использовать принципы методологии ARIS [15]. Разработ-

ка EPC-диаграмм сценариев угроз позволяет отразить на одной схеме все значимые этапы многокомпонентной атаки. Четко обозначенные события, как результаты развития атаки, позволяют отметить и настроить точки контроля индикаторов для всех значимых этапов.

EPC-диаграмма процесса реализации угрозы целевому объекту АСУ ТП может быть представлена в виде комбинации событий и функций. При этом под функцией понимаем проводимые исполнителем-киберпреступником тактики и техники. Таким образом, функция – это действие или набор действий, выполняемых в информационной среде объекта защиты киберпреступником; функция может быть поименована соответствующей техникой из приложения 11 методики [12].

EPC-диаграмма сценария угрозы представляет собой отображение (сверху вниз) последовательности выполнения техник, начиная от исходного действия киберпреступника до достижения им цели – угрозы безопасности информации (УБИ) объекту воздействия, это может быть УБИ из банка данных [16]. Реализация на определенном этапе техники или их совокупности вызывает событие – состояние информационной среды, которое может быть оценено как некоторая промежуточная УБИ, существенная для достижения объекта воздействия угрозы. Эта промежуточная УБИ оказывает влияние на дальнейшее развитие сценария. Событие отображается на диаграмме как специальный элемент. В свою очередь событие – промежуточная УБИ – активизирует последующие тактики. Функции и события в процессе реализации сценария чередуются.

Решение о развитии сценария, то есть ходе выполнения многокомпонентной атаки, принимается киберпреступником по мере поиска уязвимостей для реализации техник. Варианты этапов развития сценария следующие:

- успешное выполнение киберпреступником техники приводит одновременно к нескольким событиям, для отображения на диаграмме используется оператор AND;
- промежуточная УБИ реализуется после одновременного выполнения двух техник, для отображения используется оператор AND перед УБИ;

- очередная техника может быть выполнена только после реализации двух УБИ, это отображается с помощью оператора AND перед техникой;
- промежуточная УБИ обеспечивает возможности выполнения двух техник, используется оператор AND перед техниками;
- выполнение техники может привести к реализации одной из двух или двух УБИ, на диаграмме это обозначается с помощью оператора OR после техники и перед $УБИ_i$ и $УБИ_j$;
- УБИ может быть реализована после выполнения одной из двух или двух техник, используется оператор OR перед УБИ;
- техника сможет быть выполнена после реализации одного из двух событий, или двух одновременно, используется оператор OR перед техникой;
- выполнение какой-либо техники может привести только к одному из событий: удалось реализовать УБИ или не удалось, это обозначается на диаграмме с помощью оператора XOR перед событиями;
- УБИ реализуется в результате наличия уязвимостей для выполнения только одной из двух техник, используется оператор XOR перед УБИ;
- техника может быть выполнена сразу после реализации одной из УБИ, тогда используется оператор XOR перед техникой.

По статистике NIST [17] самыми опасными являются целенаправленные внешние атаки.

Для реализации атаки на промышленную сеть через глобальную, киберпреступник в первую очередь должен осуществить несанкционированный доступ в корпоративный сегмент и закрепиться в нем для дальнейшего проникновения за периметр промышленной сети.

Рассмотрим построение сценария атаки на корпоративный сегмент промышленного предприятия с технологической сетью, в случае, когда доступ в корпоративный сегмент (бизнес-контур) предоставляется через VPN-соединение удаленному пользователю. В этом случае уровень защищенности сети компании, в том числе технологической сети, становится зависимым от защищенности компьютера удаленного пользователя. Если должные меры безопасности не реализованы или

имеются уязвимости, киберпреступник, атаковав узел, может использовать VPN для туннелирования трафика за периметр корпоративного сегмента сети предприятия с АСУ ТП.

В результате заражения компьютера удаленного пользователя вредоносным программным обеспечением, оказываются скомпрометированы аутентификационные данные пользователя, который может иметь право доступа к каким-либо серверам бизнес-контура. Таким образом, с помощью троянской программы, которая собрала логины, пароли и другие данные, необходимые для связи с сервером бизнес-контура по каналу VPN, киберпреступник создает на своем компьютере копию скомпрометированного узла и осуществляет попытку нарушения периметра и далее подключения к серверу бизнес-контура сети, имея данные идентификации и аутентификации удаленного легитимного пользователя.

При наличии ошибок настройки прав доступа пользователей к ресурсам сервера киберпреступник после авторизации на сервере может определить директорию, в которой хранятся хэш-суммы паролей пользователей. Далее возможна процедура обратного пересчета хэш-функций, перебор собранных паролей и попытка авторизоваться на сервере под логином администратора. Права администратора позволяют ему, изменив конфигурацию сетевого оборудования, получить доступ в АСУ ТП для реализации атаки на целевой объект воздействия. На рис. 2 приведена разработанная EPC-диаграмма сценария атаки на корпоративный сегмент промышленного предприятия.

EPC-диаграмма сценария угрозы позволяет оценить численно вероятность реализации сценария и его ранг по следующему алгоритму:

$$P_K = \prod_{l=1}^{L} 0.5^n * W_{CVE_l}(T_{ij}),$$

где P_K – вероятность реализации K-го сценария;

W_{CVE_l} – нормированное значение уязвимости для реализации техники T_{ij};

n – число операторов XOR на диаграмме сценария;

$$R_K = P_K * C_m,$$

где R_K – величина риска реализации K-го сценария;

C_m – ценность ресурса (объекта воздействия угрозы).

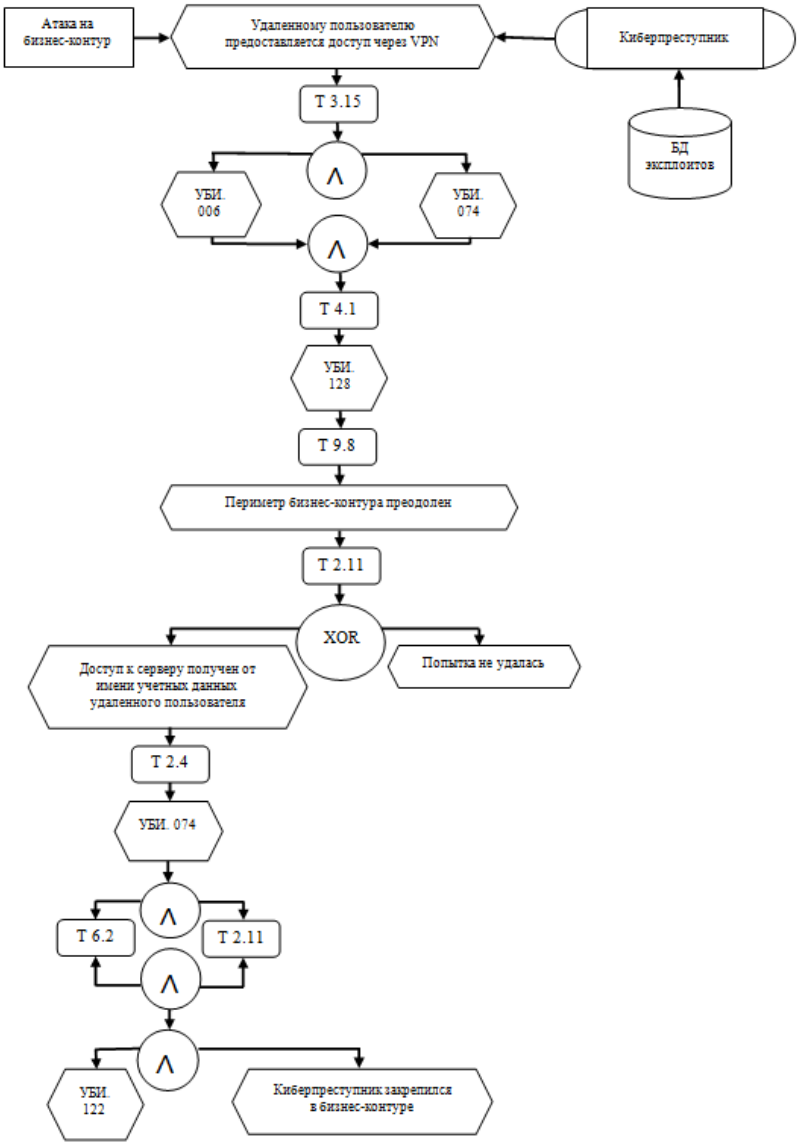


Рис. 2. EPC-диаграмма сценария атаки на корпоративный сегмент

Ранг сценария определяется в зависимости от показателя риска.

Функционирование модуля принятия решения по выбору варианта реагирования в составе архитектуры IRP связано с оценкой вероятности реализации сценария, его ранга; алгоритм должен обеспечивать минимизацию ущерба как от возможной атаки, так и от ответных действий.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Приказ ФСТЭК России от 14 марта 2014 № 31 «Об утверждении Требований к обеспечению защиты информации в автоматизированных системах управления производственными и технологическими процессами на критически важных объектах, потенциально опасных объектах, а также объектах, представляющих повышенную опасность для жизни и здоровья людей и для окружающей природной среды».
2. Приказ ФСТЭК России от 25 декабря 2017 № 239 «Об утверждении Требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации».
3. Information Security Информационная безопасность: офиц. сайт. – URL: https://www.itsec.ru/articles (дата обращения: 16.08.2024).
4. *Gaggero G.B., Armellin A., Portomauro G. and Marchese M*. Industrial Control System-Anomaly Detection Dataset (ICS-ADD) for Cyber-Physical Security Monitoring in Smart Industry Environment // IEEE Access. – 2024. – Vol. 12. – P. 64140-64149.
5. *Makrakis, Georgios Michail & Kolias, Constantinos & Kambourakis, Georgios & Rieger, Craig & Benjamin, Jacob*. Vulnerabilities and Attacks Against Industrial Control Systems and Critical Infrastructures. – 2021. – URL: https://www.researchgate.net/publication/354493711_Vulnerabilities_and_Attacks_Against_Industrial_Control_Systems_and_Critical_Infrastructures (дата обращения: 16.08.2024).
6. Jet Detective и Jet Signal внесены в реестр отечественного программного обеспечения. 12 декабря 2017. Инфосистемы Джет: офиц. сайт. – URL: https://jet.su/press-center/news/jet-detective-i-jet-signal-vneseny-v-reestr-otechestvennogo-programmnogo-obespecheniya (дата обращения: 15.08.2024).
7. Р-Вижн: офиц. сайт. – URL: https://www.rvision.ru/ (дата обращения: 15.08.2024)
8. Интеллектуальная безопасность: офиц. сайт. – URL: https://www.securityvision.ru/docs/IRP.php (дата обращения: 15.08.2024).
9. Cyberbit SOC 3D. Автоматизированная консолидация всех данных управления процессом реагирования на киберинциденты на единую панель управления и повышение эффективности SOC в целом. – URL: https://www.pacifica.kz/upload/iblock/f43/f438e9f735ad1f4218e45acb9db8d706.pdf?ysclid=lzx8mckt1r891749321 (дата обращения: 16.08.2024).

10. IBM Resilient Incident Response Platform Enterprise on Cloud delivers orchestrated and automated incident response processes. IBM Asia Pacific Software Announcement AP16-0410. November 1, 2016. – URL: https://www.ibm.com/docs/en/announcements/archive/ENUSAP16-0410#abstrx (дата обращения: 15.08.2024).
11. IRP (Incident Response Platform). – URL: https://encyclopedia.kaspersky.ru/glossary/irp/ (дата обращения: 16.08.2024).
12. Методический документ ФСТЭК России от 05.02.2021. Методика оценки угроз безопасности информации. – URL: https://normativ.kontur.ru/document?moduleId=1&documentId=451500 (дата обращения: 15.08.2024).
13. *Машкина И.В.; Гарипов И.Р.* Разработка EPC-моделей угроз нарушения информационной безопасности автоматизированной системы управления технологическими процессами // Безопасность информационных технологий, [S.I.]. – 2019. – Т. 26, № 4. – С. 6-20. – ISSN 2074-7136. – DOI: http://dx.doi.org/10.26583/bit.2019.4.01.
14. *Заид Алкилани М.О., Машкина И.В.* Разработка сценариев атак для оценки угроз нарушения информационной безопасности в промышленной сети // Проблемы информационной безопасности. Компьютерные системы. – 2024. – № 1 (58). – С. 96-109. – DOI: 10.48612/jisp/xvkx-k619-3f2z 25.04.2024. - EDN: PDNEWN.
15. *Шеер А.В.* ARIS-моделирование бизнес-процессов. – М.: Вильямс, 2000. – 175 с.
16. Банк данных угроз безопасности информации Федеральная служба по техническому и экспортному контролю России. – URL: https://bdu.fstec.ru (дата обращения: 16.08.2024).
17. *Stouffer K., Pease M., Tang CY., Zimmerman T., Pillitteri V., Lightman S., Hahn A., Saravia S., Sherule A., Thompson M.* Title. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-82r3. – 2023. – https://doi.org/10.6028/NIST.SP.800-82r3.

УДК 00. 1082

В.Д. Михайлова, Е.С. Басан

Россия, г. Таганрог, Южный федеральный университет

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПАРАМЕТРОВ АТАК И ИНЦИДЕНТОВ НА КИБЕРФИЗИЧЕСКИЕ СИСТЕМЫ

Цель работы – провести анализ параметров атак и инцидентов в киберфизических системах, а также определить как эти параметры влияют на входной и выходной трафик киберфизической системы. На основе полученного анализа можно будет выявить зависимость инцидентов от атак, а также классифицировать по анализу трафика тип атак, которые проводятся на киберфизические системы. Задачи, которые необходимо выполнить в рамках данной работы: проанализировать атаки и их параметры, которые влияют на целостность, доступность и конфиденциальность киберфизических систем; выявить параметры обнаружения атак и описать их в карточке инцидента, провести экспериментальные исследования и изучить трафик при обычной работе и во время атак на киберфизические системы.

Ключевые слова: *параметры; сравнительный анализ; атаки; инцидент; киберфизические системы.*

The objective of the work is to analyze the parameters of attacks and incidents in cyber-physical systems, and determine how these parameters affect the input and output traffic of the cyber-physical system. Based on the obtained analysis, it will be possible to identify the dependence of incidents on attacks, as well as classify the type of attacks carried out on cyber-physical systems based on traffic analysis. The tasks that need to be completed within the framework of this work: analyze attacks and their parameters that affect the integrity, availability and confidentiality of cyber-physical systems; identify attack detection parameters and describe them in the incident card, conduct experimental studies and study traffic during normal operation and during attacks on cyber-physical systems.

Keywords: *parameters; comparative analysis; attacks; incident; cyber-physical systems.*

Введение

Киберфизические системы оптимизируют множество процессов, они помогают быть человеку мобильным и автономным. Такие системы внедрены практически во все сферы жизни и профессии, их можно встретить при выращивании и сборе урожая, на заводах и предприятиях, в больницах и ресторанах, в спорте и доставке грузов, их используют в чрезвычайных ситуациях и при исследовательских задачах. Киберфизические системы отличаются тем, что они могут собирать данные от окружающей их среды и менять своё поведение на основе полученных данных. В такие системы часто внедряют искусственный интеллект, самые различные датчики и алгоритмы обработки больших данных. В данной исследовательской работе будет произведен анализ двух стендов киберфизических систем: промышленное производство (конвейерная лента) и дрон (беспилотная автоматизированная система), будет проанализирована работа таких систем, выявлены к каким атакам они чаще всего подвергаются, а также будут обнаружены причины изменения трафика в работе таких систем во время реализации атак на них.

Описание экспериментальных стендов

В работе рассматривается два стенда киберфизических систем: умное промышленное производство и беспилотная автоматизированная система рис. 1. Промышленное производство представлено в виде конвейерной ленты, которое передвигает груз и ставит на него штамп, в системе есть датчики движения – фотодиоды. Данным стендом можно управлять с помощью веб-интерфейса и передавать ему разные команды. В киберфизической системе есть система мониторинга физических параметров (загрузка ЦПУ, загрузка оперативной памяти, дискового пространства и др.) и сетевых параметров (количество пакетов различных протоколов передачи данных), а также база данных, в которую сохраняются эти данные [1]. Также в системе есть уровень автономного управления, который состоит из трех микрокомпьютеров Raspberry Pi, двух фотодатчиков и двух моторов. Один из контроллеров управляет одним мотором и одним фотодатчиком. Каждый микрокомпьютер выполняет одну из ролей:

1. Человеко-машинный интерфейс (HMI) – Интерфейс пользователя, отвечает за отображение интерфейса в веб-браузере, получение команд от пользователя и отправка их на PLC по протоколу ModBus.

2. Программно-логический контроллер (PLC) – программируемый логический контроллер – выделенный сервер управления всем технологическим процессом, взаимодействует напрямую с контроллерами, HMI и SCADA по протоколу ModBus, обрабатывает поступающие ему данные, отправляет команды управления на контроллеры и производит журналирование состояния системы (уровень обработки и аналитики).

3. SCADA – система диспетчерского мониторинга, получает данные о состоянии системы, журналирует их и отображает на интерфейсе в реальном времени (уровень обработки и аналитики).

Сетевой уровень модели представляет собой проводное подключение к маршрутизатору каждого компонента физического уровня через Ethernet – кабель. Взаимодействие между уровнями КФС происходит по протоколу промышленной автоматизации ModBus.

Беспилотная автоматизированная система также рассматривается как киберфизическая система, так как она имеет связь с наземной станцией, которая задает ей полетное задание и принимает решение по реализации необходимых действий при негативном воздействии со стороны окружающей среды или злоумышленника. С помощью данного стенда можно собирать логи, в которых отображается время и место работы стенда, количество спутников для работы системы и т.д. [2].

а б

Рис. 1. а – стенд промышленного производства, б – стенд беспилотного летательного аппарата

Разработка сценариев атак

Было разработано два сценария атак отказа в обслуживании – Обрыв связи (GPS-глушение, деаутентификация) и DoS-атаки на стенды: беспилотная автоматизированная система и промышленное производство.

DoS-атака выполняется на прикладном, транспортном, сетевом уровне и уровне представления сетевой модели. На транспортном уровне сеть нагружается большим количеством пакетов протоколов TCP или UDP, на сетевом уровне сеть нагружается пакетами протокола ICMP, на прикладном уровне пакетами HTTP и др., на уровне представления открываются SSL соединения и т.д.. Суть атаки перегрузить устройство, отправляя пакеты тем самым открывая соединения, которое не успевает обрабатывать устройство, в результате чего оно не может выполнять заданные функции. Данной атакой можно вывести из строя соединение, протокол, приложение, сеть или устройство в целом.

Существуют разновидности DoS-атак: SYN-flood и UDP-flood [3]. Например, при SYN-flood атаке отправляется большое число SYN-пакетов на открытый порт устройства, которые не приводят к действительной установке соединения из-за чего появляется большое количество «полуоткрытых соединений», которые переполняют очередь подключений, вынуждая устройство отказывать в обслуживании очередным пакетам. Реализовать атаку можно с помощью утилиты netwox [4], командой: netwox 76 -i ip-адрес жертвы -p номер порта -s raw. А также с помощью утилиты hping3, командой: hping3 -S -a ip-адрес злоумышленника --flood -p номер порта ip-адрес жертвы. Атака очень популярна и существует множество инструментов для её реализации, также не составит труда написать свой скрипт, например на языке Python для ее выполнения. Отследить реализацию атаки можно с помощью сниффера сетевого трафика Wireshark [5].

Атака GPS-глушение выполняется на физическом уровне сетевой модели, атаку можно реализовать с помощью устройств ADALM-Pluto, bladeRF, HackRF и USRP. Принцип работы данной атаки – создание большого количества ложных сигналов на рабочей частоте устройства, в результате чего устройство выходит из строя.

Экспериментальные исследования

В рамках данной работы были проведены на киберфизические системы следующие атаки: PUSH-ACK flood, UDP-flood, ACK-flood, ICMP-flood и ModBus-flood.

Практический пример атаки UDP-flood и анализ работы с последствиями атаки:

Атака выполнялась на операционной системе Linux с помощью утилиты Hping3 командой - sudo hping3 -i u150 -a 192.168.100.201 -2 -p 502 192.168.100.202 (рис. 2).

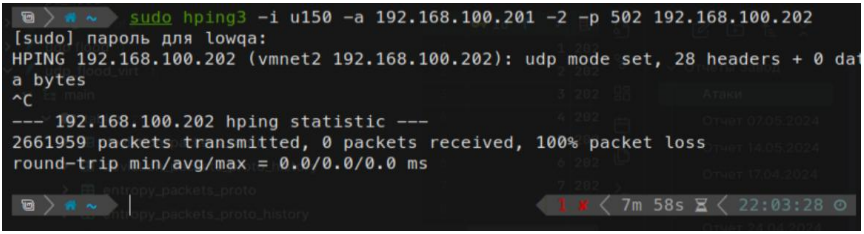


Рис. 2. Пример запуска атаки UDP-flood

Сценарий атаки – злоумышленник совершает UDP-flood атаку на 502 порт, на котором работает протокол ModBus. Он подделывает IP-адрес SCADA. Злоумышленник использует IP-адрес в качестве источника трафика и отправляет пакеты на PLC контролер. Соответственно после проделанной атаки аналитик данных ждет отклонений от нормальной работы в количестве передаваемых пакетов по протоколу UDP и в физических параметрах состоянии системы, такие как оперативная память и загрузка центрального процессора. По рис. 3 видно, что система изначально не передавала UDP пакеты и в системе было 0 пакетов, после атаки видно значительное увеличение UDP пакетов в 5 секундных интервалах и далее их снижение к 0, что означает окончание атаки. Также на рис. 4 отображена энтропия атаки. В начале виден скачок, который не относится к экспериментальному исследования работы системы во время атаки, далее на втором пике виден снова скачок, который соответствуют началу атаки и ее концу [6].

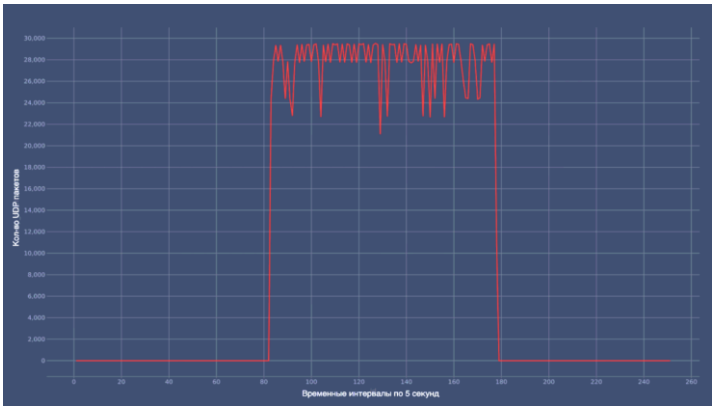


Рис. 3. Отображение кол-во пакетов в системе

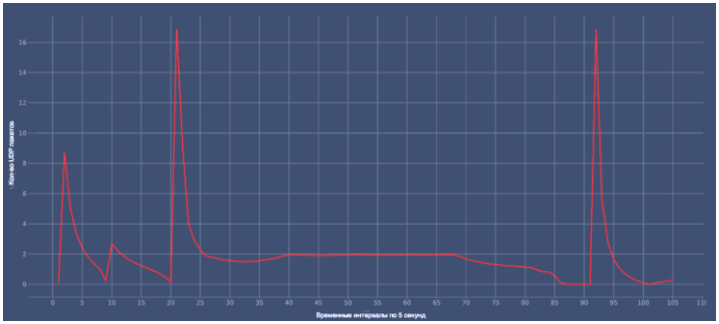


Рис. 4. Энтропия для измерения изменения числа пакетов

Также в рамках данной работы была проведена атака GPS-глушение сигнала. В атаке злоумышленник передаёт шумы с помощью направленной антенны на устройство, с целью подавления приёма сигнала GPS. В данном примере беспилотная автоматизированная система позиционируется на навигационной системе GPS.

Порядок выполнения атаки:

В программе GNU Radio необходимо построить схему, представленную на рис. 5.

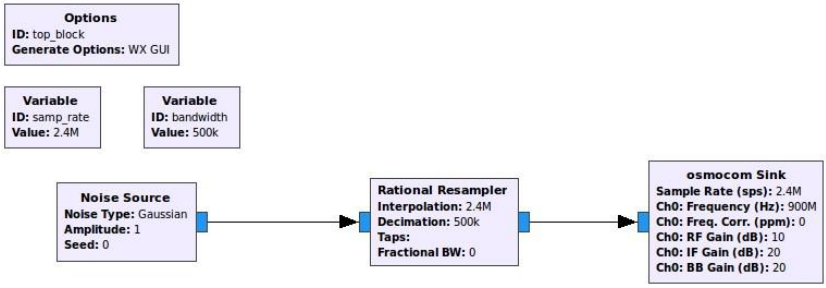


Рис. 5. Схема в программе GNU Radio

В блоке Rational Resampler необходимо указать в поле interpolation значение 1575.42M – частота на которой работают спутники. В поле decimation указать 10M – ширина канала.

Так как GPS работает в двух диапазонах, то для частоты 1227.60M и ширины канала 10M необходимо совершить аналогичные действия.

Последствия атаки:

В результате данной атаки, беспилотная автоматизированная система, выполняющая движение по маршруту с использованием позиционирования с помощью GPS, потеряло управление и совершило посадку или разбилось.

Для реализации атаки использовалось программное обеспечение GNU Radio и устройство HackRF с направленными антеннами.

После проведения атаки можно было получить с дрона журналы с логами его работы и увидеть, как изменялась количество спутников в системе [7].

Метод обнаружения атак

В рамках работы был разработан метод обнаружения атак и аномальной работы киберфизических систем. С помощью метода обнаружения атак можно анализировать нестандартное поведение сетевого трафика на устройствах и вести статистику данных [8].

Были реализованы следующие метрики:

- Нормальное распределение.
- Распределение Пуассона.
- Минимальная и максимальная нормализация.

- Отношение входящих пакетов к исходящим.
- Отношение пакетов с флагом RST ко всем остальным.
- Вероятность попадания пакетов в доверительный интервал.
- Среднее значение пакетов.
- Среднеквадратичное отклонение пакетов.

Данные метрики рассчитываются на определенных количествах интервалов времени. Было разработано приложение по сбору данных, которое считывает количество пакетов в определенный промежуток времени, после чего эти временные промежутки формируют «окно», по умолчанию окно состоит из 10 таких промежутков. После чего 10 значений (промежутков) поступают в программу, для обработки данных. При этом программа при поступлении каждого нового значения, сдвигает «окно», тем самым данные для анализа как бы смещаются, обновляясь со временем.

Рассмотрим на примере вычисления метрики среднего значения:

Установим интервал времени в 5 секунд, значение «окна» на 10, и сдвиг на 1.

Первые 5 секунд приложение будет захватывать пакеты, после чего итоговое число пакетов образует первый временной интервал, следующие 5 секунд образуется 2-рой и так далее, пока значение интервалов не достигнет установленного значения «окна» = 10. Теперь у нас есть 10 значений количества пакетов, которые мы анализируем через формулу поиска среднего значения. Последующие же значения будут накапливаться в буфере, и как только их количество достигнет параметра сдвиг = 1, мы сдвинем «окно» на установленное значение, самое же первое значение временного интервала уничтожится. Так будет продолжаться до тех пор, пока работает программа.

Проведя эксперименты, опытным путем было обнаружено, что минимальная и максимальная нормализация и нормальное распределение, хорошо обозначают аномальную работу или атаку при внезапном росте числа пакетов в сети.

В результате было принято решение объединить две метрики – сначала нормализацию данных, при помощи минимальных и максимальных значений, затем отдаем эти значения в нормальное распределение.

Для каждой группы значений получаем график, четко определяющий вероятность появления того или иного события в системе.

Также можно вычислить энтропию для каждого распределения и отображать в графике полученные значения.

Разработка схемы описания инцидентов

Инцидент информационной безопасности – это нежелательное событие, которое может привести к компрометации данных или выводу из строя информационных систем. Виды атак можно соотнести с видами инцидентов. Пример соотношений атак и инцидентов, рассматриваемых в данной работе изображено на рис. 6.

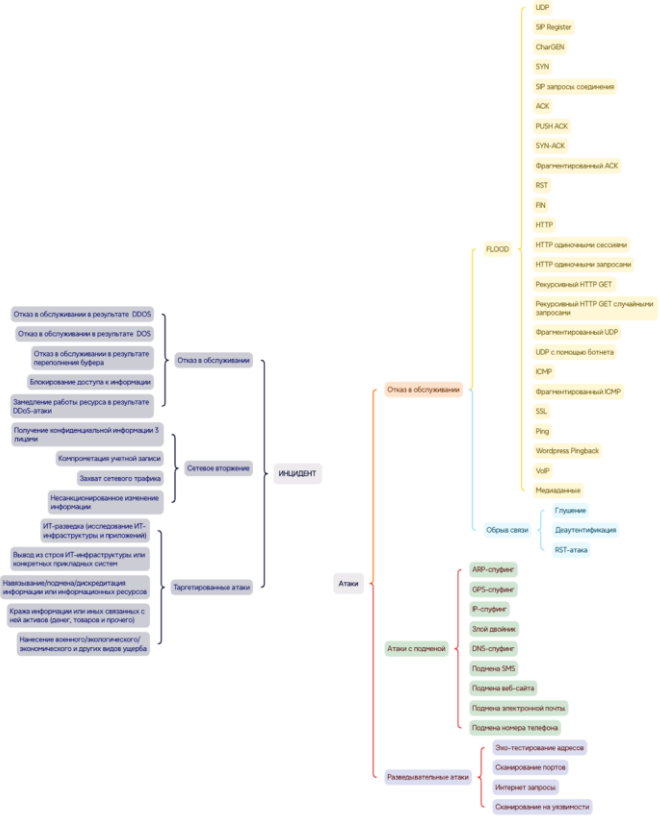


Рис. 6. Схема инцидентов для стенда

Здесь очень важно произвести сравнительный анализ параметров при реализации атак и их результата после выполнения (инцидента).

При реализации UDP-flood атаки, на жертву подается очень большое количество пакетов. В первую секунду можно обнаружить большой скачок пакетов в сетевом трафике, но далее пакетов становится меньше обычного, так как система выходит из строя. Такая же ситуация происходит при реализации атаки GPS-глушение [9].

Заключение

В заключении можно сделать вывод, что в системе можно выявить параметры, по которым можно определять наличие атак или аномальной работы системы, а также можно выявить параметры атак и инцидентов и создать прямую связь между ними. Из-за чего будет легче и быстрее выявлять атаки, которые проводятся на киберфизические системы, и формировать карточки инцидентов в отчетные документы и определять им степень критичности. В результате специалисты в сфере информационной безопасности смогу быстрее определять атаки в системе и качественней предотвращать последствия от инцидентов.

Благодарность. *Глава 1, 2 и 3 выполнены при поддержке Совета по грантам Президента Российской Федерации за счет средств стипендии Президента Российской Федерации для молодых ученых и аспирантов (Конкурс СП-2022) № СП-858.2022.5 по теме «Технология обеспечения кибербезопасности автоматизированных систем от активных информационных атак на основе принципа отражения», глава 4 и 5 выполнена при поддержке внутреннего гранта студенческим научным объединениям Южного федерального университета.*

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Basan E.S., Mikhailova V.D., Martynenko M.V*. Security Assessment of Technological Process for Smart Manufacturing // Proceedings - 2023 International Russian Automation Conference, RusAutoCon 2023. – 2023. – P. 1010-1015.
2. *Басан Е.С., Пескова О.Ю., Михайлова В.Д., Шулика М.Г.* Разработка методики анализа угроз и уязвимостей БПЛА // Информационные системы и технологии в моделировании и управлении: Сборник трудов VI Международной научно-практической конференции, Ялта, 24–26 мая 2021 года / Крымский федеральный университет имени В. И. Вернадского, Гуманитарно-педагогическая академия (филиал). – Симферополь:

Общество с ограниченной ответственностью «Издательство Типография «Ариал», 2021. – С. 235-239. – EDN HRGCWB.

3. *Basan Elena, Alexander Basan, Alexey Mushenko, Alexey Nekrasov, Colin Fidge, and Alexander Lesnikov.* Analysis of Attack Intensity on Autonomous Mobile Robots // Robotics. – 2024. – 13, No. 7: 101. – https://doi.org/10.3390/robotics13070101.
4. *Basan E., Basan A., Nekrasov A*. Method for detecting abnormal activity in a group of mobile robots // Sensors. – 2019. – 19(18):4007. – P. 1-21. – DOI: 10.3390/s19184007.
5. *Михайлова В.Д., Шулика М.Г., Басан Е.С.* Архитектура безопасности для беспилотных летательных аппаратов // Перспективные системы и задачи управления: Материалы XVI Всероссийской научно-практической конференции и XII молодежной школы-семинара, п. Нижний Архыз – п. Домбай, 05–09 апреля 2021 года. – Ростов-на-Дону: ИП Марук М.Р, 2021. – С. 31-39. – EDN NPGJWG.
6. *Басан Е.С., Михайлова В.Д., Шулика М.Г. [и др.]*. Определение набора метрик для детектирования атак на КФС // Системный синтез и прикладная синергетика: Сборник научных работ XI Всероссийской научной конференции, п. Нижний Архыз, 27 сентября – 01 2022 года. – Ростов-на-Дону – Таганрог: Южный федеральный университет, 2022. – С. 183-189. – DOI 10.18522/syssyn-2022-35. – EDN OQMHSK.
7. *Басан А.С., Басан Е.С., Иванникова Т.Н. [и др.]*. Концепция базы знаний угроз киберфизических систем на основе онтологического подхода // Системный синтез и прикладная синергетика: Сборник научных работ XI Всероссийской научной конференции, п. Нижний Архыз, 27 сентября – 01 2022 года. – Ростов-на-Дону – Таганрог: ЮФУ, 2022. – С. 172-177. – DOI: 10.18522/syssyn-2022-33. – EDN UHQKZZ.
8. *Mikhailova V.D., Shulika M.G., Basan E.S., and Peskova O.Y.*. Security architecture for UAV. Proceedings of the 2021 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), Yekaterinburg, Russia. 2021, January 13-14). – DOI: 10.1109/USBEREIT51232.2021.9455039. – EDN: DSQEHP.
9. *Basan E., Basan A., Makarevich O., Babenko L.* Studying the impact of active network attacks on a mobile robots group // Cybersecurity issues [Voprosy kiberbezopasnosti]. – 2019. – No. 1. – P. 35-44. – ISSN 2311-3456. – DOI: 10.21681/2311-3456-2019-1-35-44.

УДК 003.26

Е.Ю. Михальчук, А.Е. Боршевников, С.А. Быстревский

Россия, г. Владивосток, Дальневосточный федеральный университет

МОДЕЛЬ ОЦЕНИВАНИЯ НЕОТЛИЧИМОСТИ ЗАШИФРОВАННЫХ ДАННЫХ В ВИДЕ ТОЧЕК ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

Статья посвящена вопросу неотличимости набора данных, представленных шифрующим преобразованием в виде точек эллиптических кривых. В результате проведенных исследований была предложена модель определения неотличимости набора данных, зависимости плотности вероятности от полученного расстояния Махаланобиса между точкой эллиптической кривой и её распределением.

Ключевые слова: *эллиптическая кривая; среднее значение набора точек эллиптической кривой; ковариация; расстояние Махаланобиса; плотность вероятности.*

The article is devoted to the issue of indistinguishability of a data set presented by an encryption transformation in the form of elliptic curve points. As a result of the conducted research, a model for determining the indistinguishability of a data set, the dependence of the probability density on the obtained Mahalanobis distance between an elliptic curve point and its distribution was proposed.

Keywords*: elliptic curve; mean value of a set of elliptic curve points, covariance; Mahalanobis distance; probability density.*

В любой информационной системе использование математически обоснованных методов анонимизации повышает доверие к самой системе, гарантируя, что личности субъектов будут защищены и что обработка данных происходит честно. Раскрытие личности субъекта происходит, когда сам субъект связан с определенной записью в списке опубликованных данных. Раскрытие некоторых атрибутов субъекта происходит в случае уточнении имеющихся данных новой информацией о субъекте. Также, при взаимодействии субъектов никто не должен иметь возможность связать сообщения конкретного субъекта с ним самим, а только с несвязанным с ним псевдонимом или частным идентификатором [1].

С целью измерения различий и их математического обоснования между элементами набора данных, в данной работе будем использовать понятие оценки неотличимости зашифрованных данных. Под оценкой неотличимости будем понимать численную характеристику, полученную из определенной выборки данных. Обеспечение неотличимости зашифрованных данных может быть достигнута за счет методов k-анонимности, l-разнообразия, t-близости и e-дифференциальной приватности, имеющие строгие формальные определения. Данные считаются неотличимы, когда каждая из этих характеристик принимает определенные значения. Оценить эффективность анонимизации данных теоретически возможно используя методы анализа данных, статистические методы, а также методы теории информации, машинного обучения и других областей.

Рассмотрим значения этих свойств.

k-анонимность – свойство набора данных, которое гарантирует, что каждая запись в наборе данных неотличима от как минимум k-1 других записей. Это означает, что злоумышленник не может идентифицировать конкретную запись, зная только ее атрибуты. Оценить эффективность можно как отношение количества k-анонимных записей к общему количеству записей.

l-разнообразие – свойство набора данных, которое гарантирует, что для каждого значения связывающего атрибута существует как минимум l различных значений других атрибутов. Это означает, что злоумышленник не может определить атрибут субъекта, даже зная его точное или приближенное значение. Оценить эффективность можно как отношение количества групп k-анонимных записей, которые являются l-разнообразными, к общему количеству групп k-анонимных записей.

t-близость – свойство набора данных, которое гарантирует, что для каждого значения связывающего атрибута существует как минимум t других значений, которые находятся в пределах определенного расстояния от него. Это означает, что злоумышленник не может определить атрибут субъекта, даже зная его связывающий атрибут и некоторый другие атрибуты, которые находятся в пределах определенного расстояния от него. Оценить эффективность можно как отношение количества пар записей в наборе данных, которые являются t-близкими, к общему количеству пар записей в наборе данных.

е-дифференциальная приватность – свойство, которое гарантирует, что для двух наборов данных, отличающихся одной записью, анализ результата работы с этими наборами данных не сможет определить первоначальный набор данных. Данный метод полезен для прогнозирования более равномерного распределения вероятностей. Оценить эффективность можно как вероятность того, что злоумышленник сможет определить первоначальный набор, не более чем на e больше, чем вероятность того, что злоумышленник не сможет достичь своей цели.

Для практических криптографических систем, основанных на методах алгебраической геометрии, базовые требования выдвигаются стандартами ГОСТ 34.10-2018 и FIPS PUB 186-4 [2, 3]. Поэтому дальнейшие исследования будут рассматривать требования, выдвигаемые этими стандартами, а именно проводить вычисления в циклических подгруппах простого порядка. Представим модель измерения сходства или различий точек эллиптических кривых используя многомерное распределение для эллиптической кривой E, определенной в форме Вейерштрасса над конечным простым полем F_p, $p > 3$ – простое число: $E: y^2 = x^3 + ax + b$, где x, y – координаты точки кривой; a, b – коэффициенты кривой такие, что $4a^3 + 27b^2 \not\equiv 0 \; mod \; p$ [4]. Зависимость точек эллиптической кривой определяется через связывающее алгебраическое выражение одной точки через другую. Определить конкретное значение или силу зависимости является достаточно сложной вычислительной задачей, сравнимой по сложности с задачей дискретного логарифмирования в конечном поле. Однако, возможно установить вид зависимости и использовать полученные значения в дальнейших исследованиях.

Представим модель неотличимости зашифрованных данных используя расстояние Махаланобиса как меру расстояния между точкой эллиптической кривой и её распределением. Расстояние Махаланобиса учитывает ковариацию между элементами набора и может давать более точные результаты в многомерном случае, чем другие меры расстояния, а также расстояние не зависит от масштаба переменных [5]. Ковариация измеряет степень линейной зависимости между числовыми случайными величинами и позволяет предсказывать поведение величин [6]. Понятие ковариации над конечным по-

лем в статистическом смысле измеряет только линейную зависимость, а другие свойства или типы зависимостей в рамках групповой структуры могут существовать, но не быть обнаружены ковариацией.

Применение расстояния Махаланобиса позволяет сделать выводы о сложности проведения анализа по выявлению статистических аномалий в зашифрованном наборе данных. Расстояния Махаланобиса между точкой и её распределением с целью выявления однородности для рассматриваемого набора данных принимает следующий вид:

$$(X,Y) = \sqrt{\left(X - \hat{S}\right)^T \theta^{-1} \left(Y - \hat{S}\right)},$$

где X,Y – набор соответствующих координат рассматриваемого набора точек; $D(X,Y)$ – расстояние Махаланобиса для наборов X,Y; T – операция транспонирования; θ^{-1} – обратная ковариационная матрица; $\hat{S}[x,y]$ – координаты средней точки.

Также расстояние Махаланобиса возможно рассчитать и между каждой парой точек для измерения их сходства или различия между собой с учетом ковариации в данных. Но с целью определения соответствия всех точек единому отклонению в наборе более информативно оценивать разброс точек относительно среднего значения набора. В обеих случаях при большом значении расстояния Махаланобиса менее вероятно, что точки будут считаться неотличимыми.

Ковариационная матрица, используемая в расчете расстояния Махаланобиса, при зависимых элементах входящего набора является квадратной и симметричной относительно главной диагонали, элементы которой отображают ковариацию между точками. При этом, значения ковариаций элементов с самими собой, расположенные на главной диагонали, всегда будут положительными и равными отклонению элемента – то есть 0 [7].

Вычисление ковариации требует вычисление математического ожидания рассматриваемого набора. Обозначим за математическое ожидание набора точек на эллиптической кривой среднее значение точек набора или их ожидаемое положение. Соответственно, для некоторого набора из n точек, принадлежащих группе точек эллиптической кривой порядка q, положим среднее значение набора $\hat{S} = [n^{-1} mod\ q] \sum_{i=1}^{n} P_i$ как эквивалент умножению точки на мульти-

пликативное обратное этого числа в группе точек эллиптической кривой. Для группы точек эллиптической кривой с составным порядком всегда будет существовать не меньше одной подгруппы простого порядка. Поэтому описанный выше метод нахождения среднего значения набора точек эллиптической кривой всегда будет иметь решение с сохранением групповой операции.

Тогда для набора точек эллиптической кривой ковариация вычисляет среднее произведение отклонений координат точек от их среднего значения следующим образом:

$$cov(X,Y) = \frac{1}{n-1}\sum_{i=1}^{n}\Big(\big(X_i - \hat{S}[x]\big)\big(Y_j - \hat{S}[y]\big)\Big),$$

где $cov(X,Y)$ – функция ковариации для наборов X, Y.

При расчете статистических характеристик если мы работаем с выборкой данных, а не с полным набором, в качестве математического ожидания произведений разностей векторов со своим средним значением принято использовать поправку Бесселя $\frac{1}{n-1}$ с целью получить несмещенную оценку ковариации [8].

Опишем процедуру определения вероятности распределения точек через расстояние Махаланобиса. Плотность распределения вероятности описывает вероятность того, что входящий набор примет определенные значения относительно всех возможных значений в распределении или, значения попадут в определенный диапазон [7]. Данные неотличимы, когда они получены из одного и того же распределения или были модифицированы таким образом, что их распределение стало идентичным. То есть, если плотность распределения двух наборов сильно похожа, то можно сделать вывод о неотличимости наборов. Поэтому оценка неотличимости элементов набора может быть найдена как нормирование функции плотности вероятности набора данных:

$$U(C) = \frac{f(C)}{\int_C f(C)dC},$$

где $U(C)$ – оценка неотличимости некоторого набора C; $f(C)$ – функция плотности вероятности некоторого набора C; $\int_C f(C)dC$ – интеграл функции плотности вероятности по набору возможных значений C.

Поскольку мы рассматриваем подгруппу точек, каждая точка которой имеет ненулевую вероятность, а значение интеграла плотности вероятности по выбранной области точек эллиптической кривой будет отлично от 1, тогда функция плотности вероятности при равномерном распределении для любой точки с координатами x, y может быть определена как:

$$f(x, y) = \frac{1}{n+1}.$$

Интеграл функции плотности вероятности для подгруппы из n точек по своей области равняется площади области кривой. В данном случае константа нормирования будет определяться как:

$$z = \left(\frac{1}{n+1}\right)^q,$$

где z – константа нормирования; q – порядок эллиптической кривой.

Параметризируем эллиптическую кривую для интерпретации более простой её формы, и используем параметры полярной системы координат $x = r^2$ и $y = r^3$. Зададим область L в плоскости (r, θ), соответствующей эллиптической кривой E, как $L = \{(r, \theta) \mid 0 \leq r \leq 1, 0 \leq \theta \leq 2\pi\}$. Тогда параметр r будет являться радиусом окружности, а θ – углом между положительным направлением оси x и радиус-вектором точки на окружности [9].

Интеграл плотности вероятности набора точек определяется следующим образом:

$$\int_E f(x_1, y_1, \dots x_n, y_n) dx_1 dy_1 \dots dx_n dy_n = \left(\frac{1}{n+1}\right)^q \int_L \dots \int_L 2r \, dr d\theta,$$

где E – эллиптическая кривая; L – выбранная область.

Для большей неотличимости точек они должны быть расположены в одной области распределения. Поэтому, необходимо обеспечить функции плотности вероятности убывание по мере увеличения расстояния Махаланобиса. Подставляя полученные значения расстояния Махаланобиса в предложенную выше оценку, функция плотности вероятности для набора из n точек примет вид:

$$f(C) = \frac{1}{n+1} * D^{-n}(C).$$

Убывание функции плотности вероятности по мере увеличения расстояния Махаланобиса можно показать путем взятия производной $\frac{d}{dD} = -qD^{-n-1}$. Производная будет отрицательной для всех $D > 0$ и $n > 0$, поэтому функция будет убывающей при увеличении D. Соответственно, убывающая плотность с увеличением расстояния D может быть интерпретирована таким образом, что события (точки), находящиеся дальше от области распределения менее вероятны и тем самым могут быть отдельно идентифицированы.

Полученная оценка неотличимости U является нормированной и принимает значения в интервале [0,1], где 1 означает, что наблюдаемые точки с одинаковой вероятностью являются неотличимыми точками и случайными точками на эллиптической кривой, а 0 означает, что точки легко могут быть отличены друг от друга.

В результате проведенных исследований была предложена модель определения неотличимости набора данных, зависимости плотности вероятности от полученного расстояния Махаланобиса между точкой эллиптической кривой и её распределением. Полученная зависимость может быть использована для оценки схожести или различия зашифрованных данных, представленных в виде точек эллиптической кривой. Предложенная модель может быть расширена с учетом подобранных к конкретной задаче статистических характеристик и учитывать дополнительные данные криптосистемы и внешние факторы, оказывающие влияние на принимаемые значения элементов в наборах или характеристики.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Нестеренко А.Ю., Семенов А.М*. Методика оценки безопасности криптографических протоколов //Прикладная дискретная математика. – 2022. – №. 56. – С. 33-82.
2. ГОСТ 34.10-2018. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. //Москва. Стандартинформ. – 2018. – С. 6-8.
3. Federal Information Processing Standards Publication (FIPS PUB) 186-4. Digital Signature Standard. //Gaithersburg: National Institute of Standards and Technology. – 2013. – P. 26.
4. *Бессалов А.В.* Эллиптические кривые в форме Эдвардса и криптография. //Монография. – 2017. – С. 8-25.

5. *Ghorbani H*. Mahalanobis distance and its application for detecting multivariate outliers //Facta Universitatis. Mathematics and Informatics. Vol 34. № 3. – 2019. – С. 583-595.
6. *Хацкевич В.Л*. Средние, квазискалярное произведение и ковариация нечетких чисел //Актуальные проблемы прикладной математики, информатики и механики. – 2021. – С. 1147-1151.
7. *Добронец, Б.С.* Вычислительный вероятностный анализ: модели и методы: монография. Сибирский федеральный университет. – 2020. – С. 34-40.
8. *Radziwill N.M.* Statistics (the easier way) with R. // Lapis Lucera. – 2017. – P. 29-33.
9. *Обухов В.А.* Криптография на основе эллиптических кривых (ECC) // Ферганский филиал Ташкентского университета информационных технологий имени Мухаммада Ал-Хорезми. – 2023. – Т. 1, № 4. – С. 182-188.
10. *Nekovar J.* Elliptic functions and elliptic curves. – 2004. – P. 12-18.
11. *Каменко Д.А., Гундина М.А., Жданович М.Н.* Особенности определния аномальных значений в системе Wolfram Mathematica // Информационные системы и технологии. – 2022. – С. 170-175.
12. *Болотов А.А.*, *Гашков С.Б., Фролов А.Б.* Элементарное введение в эллиптическую криптографию. Протоколы криптографии на эллиптических кривых: монография. Изд. № 4. – 2012.

УДК 004

Г.С. Омаров, Р.Ж. Сатыбалдиева, А.А. Фесенко

Республика Казахстан, г. Алматы, Казахский национальный исследовательский технический университет им. К.И. Сатпаева

АЛЬТЕРНАТИВНЫЙ МЕТОД УПРАВЛЕНИЯ КРИПТОВАЛЮТНЫМИ КОШЕЛЬКАМИ

Для мобильного управления криптовалютными кошельками разрабатываются горячие кошельки. Рассматривается возможность управлять блокчейн кошельками с помощью месенджера телеграм. В данной статье также рассматривается возможность применения данного метода.

Ключевые слова: *блокчейн; блокчейн кошелек; криптовалюта; телеграм месенджер; мобильные приложения; bitcoin (btc); litecoin (ltc); KZCash (kzc); Qiwi; API.*

Hot wallets are being developed for mobile management of cryptocurrency wallets. The possibility of managing blockchain wallets using the Telegram messenger is being considered. This article also discusses the possibility of using this method.

Keywords: *blockchain; blockchain wallet; cryptocurrency; Telegram messenger; mobile applications; bitcoin (btc); litecoin (ltc); KZCash (kzc); Qiwi; API.*

Введение

Блокчейн – это не просто объект, продукт, тенденция или некая возможность. Блокчейн состоит из нескольких частей, некоторые из которых работают вместе, а другие – самостоятельно и независимо. Благодаря этой модульности блокчейн имеет бесконечное множество вариантов использования. В долгосрочной перспективе большинство пользователей не будут знать или понимать, что в программном обеспечении или сервисе, которым они пользуются, присутствует блокчейн [1].

Блокчейн – это децентрализованный журнал записи транзакций, который является частью более широкой вычислительной инфраструктуры, которая также должна включать в себя много других функций [2].

Цель

В связи с тенденцией развития криптовалютной индустрии, для любой криптовалюты необходимо разработать горячие кошельки. Это обычно онлайн кошелек, который работает через определенный интернет сайт или мобильное приложение разработанное для часто используемых операционных систем.

В последнее время участились мошеннические действия при обмене в онлайн криптовалюту. И не знающие тонкостей блокчейна часто попадаются на такие случаи. В связи с этим данный телеграм кошелек также должен обеспечивать безопасную и гарантированный обмен фиатных денег на криптовалюту.

Целью данной работы является - выбор и реализация, относительно дешёвого но в тоже время надежного метода управления криптовалютным кошельком в онлайне.

Методы

Первый самый простой метод предоставления доступа к управлению блойчейн кошельком – это создание Web сайта. Этот метод можно использовать когда нужно быстро реализовать. Но при использовании данного метода, надо защитить сайт, от разного рода атак начиная DOS и DDOS атак.

Второй метод – это разработка мобильного приложения. Использование данного метода требует разработку для каждой операционной системы отдельно. Также нужна публикация на ресурсах как Плэймаркет, Аппстор.

Третий метод, который был использован нами – это в качестве приложения использовать мессенджер телеграм. Данный метод может покрыть недостатки первого и второго метода. Нет необходимости разрабатывать и сопровождать мобильное приложение, это обеспечивает мессенджер телеграм.

Реализация задачи

В первую очередь Вам необходимо создать телеграм бота с помощью отца ботов – https://t.me/BotFather .

Второй основной задачей является обеспечение сторону бэкэнд. Куда телеграм будет отправляет запросы пользователя и получать на них ответы. Так как нам нужно обеспечить работу телеграм

бота 24/7 и не требующих вложений на приобретение сервера, было решено арендовать виртуальный сервер (далее VPS) на облачных сервисах [3].

Нужен VPS, где мы могли бы установить использовать операционную систему Linux Ubuntu. Искали VPS с объемом оперативной памяти не ниже 4 ГБ и постоянная память на носителе SSD объемом не ниже 50 Гб [4].

Один из ведущих компании Казахстана для аренды виртуальных серверов компания ps.kz предлагает следующие решения (рис. 1).

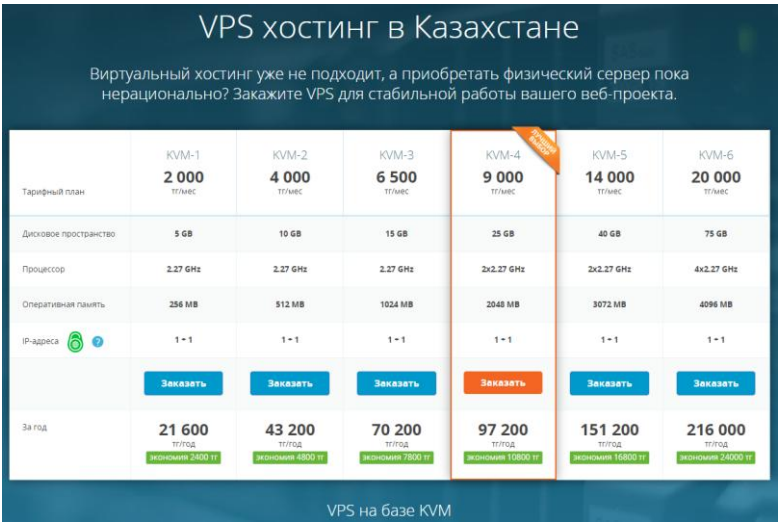


Рис. 1. Решения компании PS.kz

Из предложенных решений нам подходит только последний «KVM-6» за 20 000 тенге в месяц. Мы считаем, что это будет дорого. Поэтому решили искать альтернативные решения за рубежом. Для нас скорость интернета будет быстрым до европейских серверов по сравнению с американскими. Рассмотрели решения европейских компаний, в частности компании Aruba из Италии, которая предлагает такие решения (рис. 2).

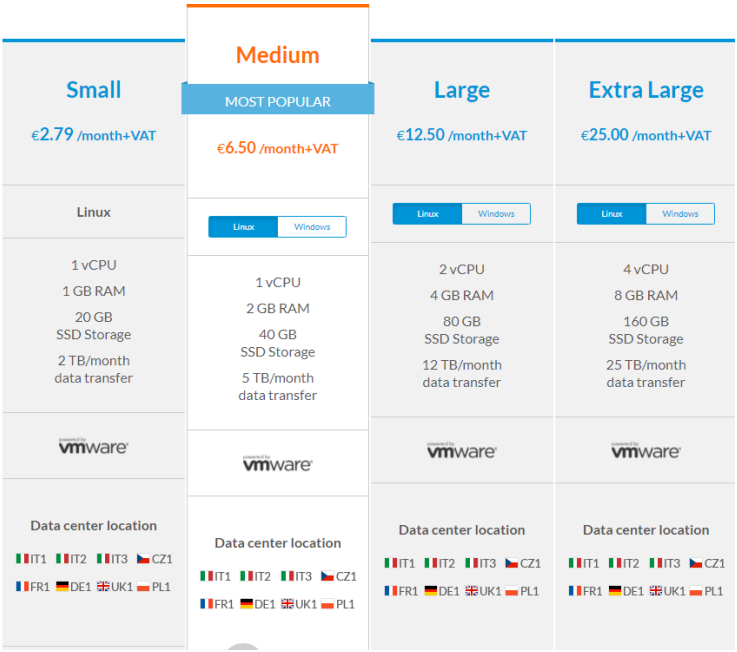


Рис. 2. Решения компании Aruba, Италия

Из предложенных, наших требований покрывает решение «Large» за 12.50 Евро, в эквиваленте около 6500 тенге.

Далее рассмотрели еще одну из крупных компаний Германии «Contabo». Там были предложены такие решения (рис. 3).

Здесь на левой стороне решения на HDD разогнанные, а с правой стороны на реальном SSD. Из них наших требований покрывает первое решение «VPS S SSD». Стоимость данного решения 4.99 Евро, в эквиваленте около 2600 тенге. Мы остановились на этом решении. Данный сервер с 4-х ядерный виртуальным процессором 2.2 GHz, с объемом оперативной памяти 8 Гб, и объем SSD диска размером 200 Гб.

На данный сервер была установлена операционная система Linux Ubuntu 16.04 (рис. 4).

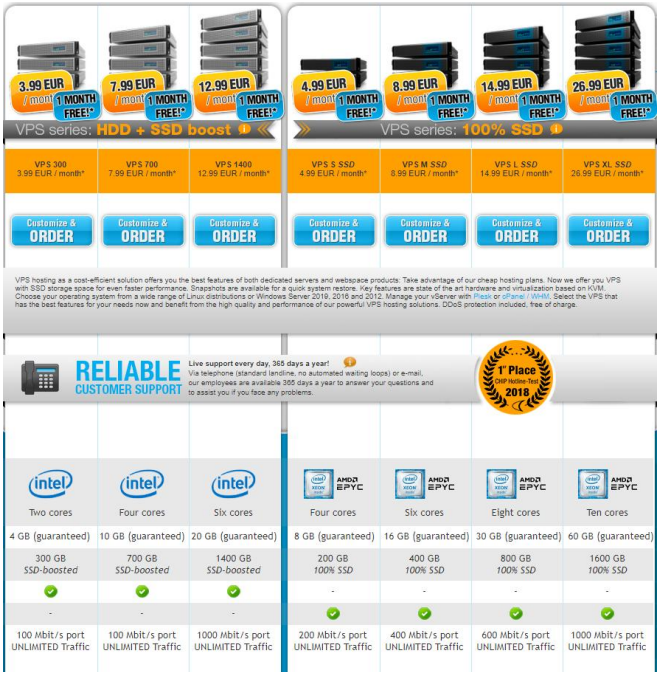


Рис. 3. Решения компании Contabo, Германия

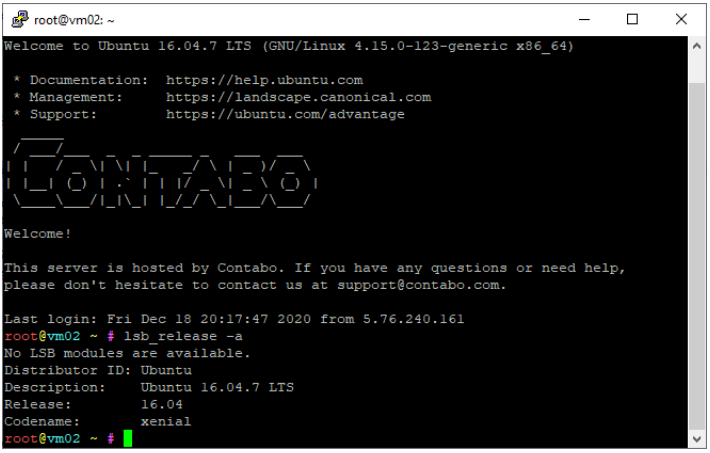


Рис. 4. Сервер с установленной ОС

В качестве языка разработки сервиса был выбран язык Python. На данный сервер была установлена среда разработки языка Python. Этот язык не был выбран случайно. Для языка Python есть готовые библиотеки для управления блокчейн кошельками, и с его помощью удобно обращаться с помощью API разным сервисам. Телеграм бот должен будет обмениваться разными сервисами через API [5].

На сервере были установлены блокчейн кошельки криптовалют – KZ Cash, Bitcoin, Litecoin. Для установки достаточно выполнить следующие команды в командной строке ОС Linux [6]:

```
# wget https://raw.githubusercontent.com/kzcashteam/mn_install/master/kzcash_mn_install.sh
# chmod +x kzcash_mn_install.sh
# ./kzcash_mn_install.sh

# wget https://bitcoincore.org/bin/bitcoin-core-0.20.1/bitcoin-0.20.1-x86_64-linux-gnu.tar.gz
# tar xvf bitcoin-0.20.1-x86_64-linux-gnu.tar.gz

# wget https://download.litecoin.org/litecoin-0.17.1/linux/litecoin-0.17.1-x86_64-linux-gnu.tar.gz
# tar xvf litecoin-0.17.1-x86_64-linux-gnu.tar.gz
```

После установки надо настроить в конфигурационном файле каждой монеты порты с которыми они будут работать. Для монеты KZ Cash файл настроек будет следующими данными:

```
rpcuser={имя пользователя kzc}
rpcpassword={пароль пользователя kzc}
rpcport=8279
listen=1
server=1
rpcallowip=127.0.0.1
```

Для Bitcoin:

```
rpcuser={имя пользователя btc}
rpcpassword={пароль пользователя btc}
rpcport=9341
listen=1
server=1
rpcallowip=127.0.0.1
```

Для Litecoin:

```
rpcuser={имя пользователя ltc}
rpcpassword={пароль пользователя ltc}
rpcallowip=127.0.0.1
rpcport=9332
listen=1
server=1
```

После внесения изменения данных в файлах конфигураций, необходимо запустить демоны кошельков:

```
# kzcashd -daemon
# bitcoind –daemon
# litecoind –daemon
```

Теперь переходим на сторону настроек Python. Надо создать там тоже конфигурационный файл, который поможет обращаться кошелькам. Пример конфигурационного файла:

```
wallet_host = "127.0.0.1"
wallet_port_kzc = 8279
wallet_user_kzc = {имя пользователя kzc}
wallet_passwd_kzc = {пароль пользователя kzc}
wallet_url_kzc="http://{user}:{passwd}@{host}:{port}".format(user=wal
let_user_kzc, passwd=wallet_passwd_kzc, host=wallet_host, port$

wallet_port = 9341
wallet_user = {имя пользователя btc}
wallet_passwd = {пароль пользователя btc}
wallet_url="http://{user}:{passwd}@{host}:{port}".format(user=wallet_u
ser, passwd=wallet_passwd, host=wallet_host, port=wallet_port)

wallet_port_ltc = 9332
wallet_user_ltc = {имя пользователя ltc}
wallet_passwd_ltc = {пароль пользователя ltc}
wallet_url_ltc="http://{user}:{passwd}@{host}:{port}".format(user=wall
et_user_ltc, passwd=wallet_passwd_ltc, host=wallet_host, port$
```

Ниже пример кода, как можно обращаться криптовалютному блокчейн кошельку с помощью Python [7–9]:

```
from bitcoinrpc.authproxy import AuthServiceProxy, JSONRPCException
from config import wallet_url, wallet_url_kzc, wallet_url_ltc

def main(coin, wallet_addr):
  if coin == 'kzc':
```

```
            cur_url = wallet_url_kzc
    elif coin == 'ltc':
            cur_url = wallet_url_ltc
    else:
            cur_url = wallet_url

    acc = AuthServiceProxy(cur_url)
    unspent = acc.listunspent(0)
    for i in unspent:
            if i["address"] == wallet_addr:
                    if count >= amount_fee:
                            break
                    txid_vout.append({"txid":i["txid"], "vout":i["vout"]})
                    count += i["amount"]

if __name__ == "__main__":
    main()
```

Необходимо использовать базу данных для учета определенных данных, для этого был выбран СУБД MySQL. Данное СУБД легкое в использовании и обеспечивает работу нашего сервиса [10].

Для установки СУБД MySQL, необходимо выполнить следующую команду:

```
# apt-get install mysql-server
```

Чтобы подключиться к базе из Python, необходимо подключить соответствующую библиотеку и с его помощью выполнять манипуляции [11]:

```
import pymysql
        class sql_db:
            def __init__(self):
                    try:
                            # устанавливаем подключение к базе
данных
                            self.conn = pymysql.connect(
                                    unix_socket=config.ms_unix_socket,
                                    user=config.ms_db_user,
                                    passwd=config.ms_db_passwd,
                                    db=config.ms_db_name,
                                    use_unicode=True, charset='utf8'
                                    )
```

```
                except pymysql.OperationalError as e:
                        print("can't connect to Mysql")
                        print(e)
                try:
                        self.cursor = self.conn.cursor()
                        self.cursor.execute('SET autocommit = 0;')
                except pymysql.OperationalError as e:
                        print("can't get cursor")
                        print.error(e)


        def select_user_id(self, user_id):
                try:
                        self.cursor.execute("SELECT * FROM table1
where id = '{user_id}'".format(user_id=user_id))
                except pymysql.Error as e:
                        print("select_user_id, error")
                        print(e)
                        return
                result = self.cursor.fetchall()
                if len(result) <= 0:
                        print("select_user_id,      error,      len(res)=0,
user_id={user_id}".format(user_id=user_id))
                        return False
                else: return result[0]
```

Далее нужно подключиться к финансовой системе, для этого была выбран сервис Qiwi. Преимущество данного сервиса в том что им можно пользоваться как безналичным так и наличным способом с помощью терминалов Qiwi. Также данный сервис предоставляет доступ к своему счету через API. Пример программного кода из Python [12]:

```
import requests
import json

def main():
  s = requests.Session()

  s_tok = "токен"
  s_log = "логин"
```

```
        s = requests.Session()
        s.headers['authorization'] = 'Bearer ' + s_tok
        parameters = {'rows': '10', 'operation': 'IN'}
        h=s.get('https://edge.qiwi.com/payment-
history/v1/persons/'+s_log+'/payments', params = parameters)
        r_str=json.loads(h.text)
        print(r_str)
```

Для подключения к банкам не удалось получить API для доступа к своему счету, и поэтому был реализован через сторонний сервис – Дзен мани. Это сервис для ведения домашней бухгалтерии. В нем есть возможность подключиться к многим банкам второго уровня Казахстана. И у данного сервиса есть доступ к своему аккаунту через API [13–15].

Пример кода обращения к сервису Дзен мани:

```
    s_tok = "токен"
    dt = datetime.datetime.now()
    timestamp = time.mktime(dt.timetuple())

    s = requests.Session()
    s_servtime = ""
    s.headers['Authorization'] = 'Bearer ' + s_tok
    logger.info("currentClientTimestamp={0},
serverTimestamp={1}".format(timestamp, s_servtime))

    parameters = {"currentClientTimestamp": timestamp,
"serverTimestamp": float(s_servtime)}
    h = s.post("http://api.zenmoney.ru/v8/diff", json = parameters)
    r_str=json.loads(h.text)
    print(r_str)
```

Рабочий вариант телеграм бота сейчас доступен по этому адресу - https://t.me/KZCashBot

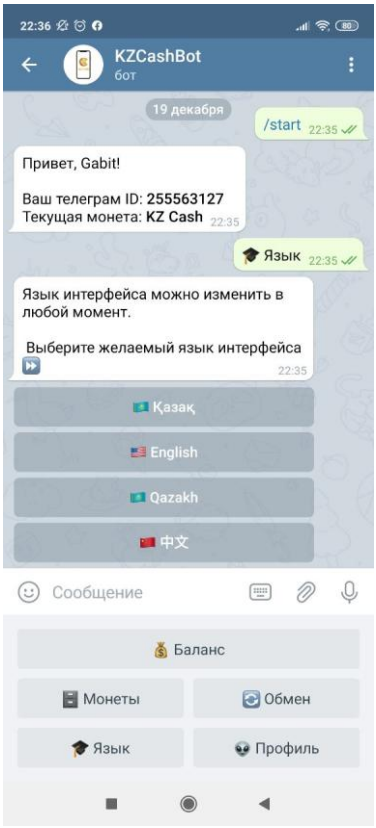


Рис. 5. Стартовая страница телеграм бота и основное меню

Рис. 6. Меню добавленных монет и меню для обмена криптовалют

Заключение

В данной статье были рассмотрены известные методы управления горячими (онлайн) кошельками. Нужен был оптимальный вариант, который будет реализован быстро, поддерживается на всех платформах, не сложная поддержка и с наименьшими расходами.

В результате был выбран оптимальный вариант реализации через месенджер телеграм. Также для оптимизации расходов на серверное оборудование был арендован VPS на облачных сервисах.

Стоимость аренды VPS на 1 месяц составляет 4.99 EUR. На реализацию данной задачи и до ввода в промышленную эксплуатацию было потрачено около 40 дней одного человека.

Данный кошелек также был интегрирован фиатными деньгами через платежную систему Qiwi. Имеется возможность частично интегрировать другими банками второго уровня с использованием сервиса Дзен-мани.

В планируется еще добавить в данный телеграм бот – токены на смартконтрактах и также сервис по запуску и управлению мастернод.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Лоран Лелу.* Блокчейн от А до Я. Все о технологии десятилетия. – Изд-во «Эксмо», 2018.
2. Nakamoto, Satoshi (31 October 2008). "Bitcoin: A Peer-to-Peer Electronic Cash System".
3. Интернет ресурс – Пишем ботов для Telegram на языке Python. – https://groosha.gitbook.io/telegram-bot-lessons/.
4. *Котенок Д.* Ubuntu Linux: базовый курс. – Брянск, 2009.
5. Интернет ресурс – Bitcoin Developer Reference. – https://bitcoin.org/en/developer-reference.
6. *Колисниченко Д.Н.* Linux - Полное руководство. – СПб.: Наука и техника, 2006.
7. Интернет ресурс – Создание RAW транзакций Bitcoin. – https://bits.media/raw_transactions/.
8. Интернет ресурс – Объяснение блокчейна для веб-разработчиков, https://habr.com/ru/post/323128/.
9. Интернет ресурс – API reference (JSON-RPC). – https://en.bitcoin.it/wiki/API_reference_(JSON-RPC).
10. Интернет ресурс – Руководство по MySQL. – https://metanit.com/sql/mysql/.
11. Интернет ресурс – Python 3 для начинающих. – https://pythonworld.ru.
12. Интернет ресурс – Руководство API Qiwi кошелька. – https://developer.qiwi.com/ru/qiwi-wallet-personal/#intro.
13. Интернет ресурс – Делаем приём платежей криптовалютой своими руками. – https://habr.com/ru/post/350430/.
14. Интернет ресурс – Telegram бот через webhook. – https://retifrav.github.io/blog/2018/12/02/telegram-bot-webhook-ru./.
15. Интернет ресурс – Программный интерфейс Дзен-Мани. – http://developers.zenmoney.ru/index.html#!/index/.

УДК 511.5+004

В.О. Осипян, Э.Т. Альгариб, А.С. Жук, К.И. Литвинов

Россия, г. Краснодар, Кубанский государственный университет

РАЗРАБОТКА КОДОВЫХ СИСТЕМ НА ОСНОВЕ ЛИНЕЙНЫХ ДИОФАНТОВЫХ СРАВНЕНИЙ

Посвящается светлой памяти просветителя науки, человека большой души Олега Борисовича Макаревича

В работе рассматриваются линейные диофантовы сравнения специального вида над кольцом вычетов, и изучается механизм нахождения всех решений указанных сравнений для построения линейных кодовых систем с заранее заданными свойствами. В частности, предлагается способ построения хорошо известного в литературе кода Варшамова.

Ключевые слова: *кольцо вычетов; поле Галуа; неопределенное уравнение; линейное диофантово сравнение; линейный код; линейный код на четность; код Варшамова.*

The paper considers linear Diophantine congruences of a special type over a residue ring, and studies the mechanism for finding all solutions of these congruences for constructing linear code systems with predetermined properties. In particular, a method for constructing the Varshamov code, which is well known in the literature, is proposed.

Keywords: *residue ring; Galois field; indefinite equation; linear Diophantine congruence; linear code; linear parity code; Varshamov code.*

I. Введение

Одной из важных современных проблем общей теории связи является повышение надежности передачи сообщений, а именно обнаружение и исправление канальных ошибок.

Как известно [1–3], сравнения и математические методы на их основе играют важную роль в современной ИКТ. Системы защиты информации на основе сравнений могут использовать простые и эффективные методы для защиты данных в целом. Эти методы часто используются как часть более сложных криптографических протоколов и систем безопасности на основе модульной арифметики.

В первой части работы приведены основные понятия, факты и определения из теории диофантова анализа [1–3], используемые нами при построении линейных кодовых систем.

Во второй части работы представлены решения специальных линейных диофантовых сравнений и соответствующие им кодовые системы, в частности, предложен способ построения хорошо известного в литературе кода Варшамова [7, 8].

II. Линейные диофантовы сравнения и простейшие кодовые системы

Пусть *a, b* целые, а *m* натуральное, отличное от единицы, числа: говорят, что *a* сравнимо с *b* по модулю *m*, если разность *a – b* делится на *m.* Данный факт записывают как $a \equiv b \ (mod\ m)$. На практике чаще записывают как $a = b \ (mod\ m)$.

Предварительно рассмотрим упрощенный вариант криптографической системы, использующей сравнение для проверки аутентичности сообщения [10]. Пусть имеется система, которая проверяет корректность переданных данных с помощью линейного сравнения:

$$c \equiv a + b \cdot k \ (mod\ m), \qquad (1)$$

где *a, b, m* – известные параметры *k* – секретный ключ, а *c* – проверочное значение, передаваемое вместе с сообщением.

Для проверки подлинности сообщения нужно проверить, что полученное значение *c* удовлетворяет сравнению (1) для известного *a* и *b*. Если сравнение (1) верно, данные считаются подлинными, иначе – возможно, произошла ошибка или атака.

Как известно [1, 2, 9], в теории линейных сравнений предметом исследования являются неопределенные (или диофантовы) уравнения первой степени с целыми коэффициентами вида:

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = a.$$

В настоящей статье мы изучаем линейные диофантовы сравнения (ЛДС) в кольце вычетов Z_m с некоторыми ограничениями вида:

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = a \ (\mathrm{mod}\ m), \qquad (2)$$

где *a* – произвольное целое, удовлетворяющее соотношению $0 \le a < m$, $m > n$,

$$a_1, a_2, \ldots, a_n \in Z,\ a_1 * a_2 * \ldots * a_n \neq 0.$$

Сравнение (2) можно записать ещё в виде:

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \overset{m}{=} a.$$

Обозначим через $N(a_1x_1 + a_2x_2 + \ldots + a_nx_n \overset{\mathbf{m}}{=} a)$ – количество решений сравнения (2) над Z_m, и отметим, что оно не зависит от величины a, поэтому будем считать его равным нулю, а при вычислениях для удобства опускать и модуль m.

Предварительно рассмотрим некоторые отдельные случаи относительно значения модуля m.

1. Пусть для кольца вычетов Z_m с обычными арифметическими операциями по модулю m, число m – простое, обозначаемое через p.

Рассмотрим линейное диофантово сравнение от n переменных над Zp, считая, что $0 \le a < p$:

$$x_1 + x_2 + \ldots + x_n \overset{\mathbf{p}}{=} a. \qquad (3)$$

Для $a = 0$ перепишем сравнение (3) в виде:

$$x_1 \overset{\mathbf{p}}{=} (p-1)(x_2 + \ldots + x_n).$$

Здесь, для каждого из

$$N(x_2 + \ldots + x_n \overset{\mathbf{p}}{=} 0) = p^{n-1}$$

p-ичного набора $(\alpha_2, \ldots, \alpha_n)$, $\alpha_i \in Zp$, $i = 2 \ldots n$, длины $n - 1$, однозначно определяется соответствующее значение α_1 для x_1 как

$$\alpha_1 = (p-1)(\alpha_2 + \ldots + \alpha_n). \qquad (4)$$

Все решения сравнения (3) образует код, содержащий p^{n-1} различных слов (каждому решению сравнения соответствует одно кодовое слово).

Соотношение (4) – это так называемое проверочное соотношение линейного p-ичного кода $C_{[n,\ n-1]}$ проверки на чётность. Этот код обнаруживает одну симметричную канальную ошибку [4-6].

Приведём следующие примеры относительно параметров n, p и a. Так, например, при $n = 4$, $p = 2$, $a = 0$, из (3) имеем:

$$x_1 + x_2 + x_3 + x_4 = 0 \ (mod\ 2),$$

все решения которого образует линейный $C_{[4,\ 3]}$ код проверки на чётность, содержащий $N(x_1 + x_2 + x_3 + x_4 = 0) = 2^3$ различных кодовых слов, представленные в табл. 1:

Таблица 1

Кодовые слова двоичного линейного кода $C_{[4,\ 3]}$

0000	*1001*	*1010*	*0011*	*1100*	*0101*	*0110*	*1111*

Как видно из Таблицы 1 все кодовые слова содержит чётное число единиц – откуда и название данного кода.

Для $n = 3$, $p = 3$, $a = 1$ имеем сравнение:

$$x_1 + x_2 + x_3 = 1 (mod\ 3),$$

имеющее $N(x_1 + x_2 + x_3 = 1) = 3^2$ решений, которые являются кодовыми словами троичного обобщённого квазилинейного кода $C_{(3,\ 2)}$, представленный в табл. 2.

Таблица 2

Кодовые слова троичного обобщённого квазилинейного кода $C_{(3,\ 2)}$

001	*010*	*220*	*100*	*211*	*121*	*202*	*112*	*022*

Очевидно, для нахождения всех решений сравнения (3) в общем случае необходимо представить его, например, в виде:

$$x_1 \overset{p}{=} a - (x_2 + \ldots + x_n)$$

и рассмотреть все наборы для правой части с определением – левой.

2. Теперь рассмотрим ещё одно ЛДС над Z_m вида:

$$x_1 + 2x_2 + \ldots + nx_n \overset{m}{=} a, \qquad (5)$$

считая, что $n < m$, $0 \le a < m$.

Предварительно рассмотрим отдельные частные случаи относительно параметров *a, m, n* сравнения (5).

Так, например, при $n = 3, m = 4, a = 0$ имеем сравнение

$$x_1 + 2x_2 + 3x_3 = 0 \ (mod\ 4),$$

и следующие $N(x_1 + 2x_2 + 3x_3 = 0) = 4^2$ решения – кодовые слова 4-ичного линейного $C_{[3,\ 2]}$-кода, представленный в табл. 3.

Таблица 3

Кодовые слова 4-ичного линейного кода $C_{[3,\ 2]}$

000	*101*	*202*	*303*	*210*	*311*	*012*	*113*
020	*121*	*222*	*323*	*230*	*331*	*032*	*133*

При $n = 3, m = 4, a = 1$ имеем сравнение $x_1 + 2x_2 + 3x_3 = 1 \ (mod\ 4)$, и следующие $N(x_1 + 2x_2 + 3x_3 = 1) = 4^2$ решения – кодовые слова 4-ичного квазилинейного кода $C_{(3,\ 2)}$, представленный в табл. 4.

Таблица 4

Кодовые слова 4-ичного обобщённого квазилинейного кода $C_{(3,\ 2)}$

100	*201*	*302*	*003*	*310*	*011*	*112*	*213*
120	*221*	*322*	*023*	*330*	*031*	*132*	*233*

В заключении приведём процедуру, позволяющая упростить определение всех корней произвольного ЛДС над Z_p.

Пусть необходимо найти все решение заданного ЛДС над Z_p от *n* переменных:

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \overset{p}{=} a \qquad (4)$$

считая, что $0 \le a < p$, $a_1, a_2, \ldots, a_n \in Z$, $a_1 * a_2 * \ldots * a_n \neq 0$.

Можно считать, что хотя бы один из коэффициентов a_i, $i = 1..n$ сравнения (4), например, $a_1 = 1$. В самом деле, т. к. $(a_1, p) = 1$, то можно найти для a_1 число $a_1' = a_1^{-1}$, так что $a_1a_1' = 1 \ (mod\ p)$, и тогда сравнение (4) можно переписать в эквивалентной форме

$$a_1x_1 + a_2\, a_1a_1'x_2 + \ldots + a_n\, a_1a_1'x_n \overset{p}{=} a_1a_1'a,$$

или

$$a_1 (x_1 + a_2 a_1' x_2 + \ldots + a_n a_1' x_n) \overset{p}{=} a_1 a_1' a$$

или

$$x_1 + a_2' x_2 + \ldots + a_n' x_n \overset{p}{=} a',$$

где $a_i' = a_i a_1'$, $a' = a_1' a$.

В завершении ещё раз вернёмся к линейному диофантовому сравнению в кольце вычетов Z_m при $m = n + 1$:

$$x_1 + 2x_2 + \ldots + nx_n = a \ (mod\ m). \qquad (5)$$

Решения сравнения (5) задают известий код Варшамова [7, 8], частные примеры которых приведены выше в табл. 3 и 4 для $n = 3$, $m = 4$.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Граве Д.А.* Элементарной курс теории чисел. – 2-е изд. – Киев, 1913.
2. *Егоров Д.Ф.* Элементы теории чисел. – Москва-Петроград, 1923. – 200 с.
3. *Лидл Р., Нидеррайтер Г.* Конечные поля. – М.: Мир, 1988. – 820 с.
4. *Мак-Вильянс Ф.Дж., Слоэн Н.Дж.А.* Теория кодов, исправляющих ошибки: пер. с англ. / под ред. Л.А. Бассалыго. – М.: Связь, 1979. – 744 с.
5. *Биркгоф Г., Барти Т.* Современная прикладная алгебра: пер. с англ. / под. ред. Ю.И. Манина. – М.: Мир, 1976. – 400 с.
6. *Блейхут Р.* Теория и практика кодов, контролирующих ошибки: пер. с англ. / под. ред. К.Ш. Зигангирова. – М.: Мир, 1986. – 576 с.
7. *Варшамов Р.Р.* О некоторых особенностях линейных кодов, корректирующих несимметрические ошибки // Докл. АН СССР. – 1964. – Т. 157, № 3. – 3 с.
8. *Варшамов Р.Р., Тененгольц Г.М.* Код, исправляющий одиночные несимметрические ошибки. // Автоматика и телемеханика. – 1965. – № 2. – 5 с.
9. *Аванесов Э.Т.* Оценка числа решений линейного диофантова уравнения // Тр. ин-та / Ивановский гос. пед. ин-т.– 1963. – Т. 34. – С. 3-7.
10. *Осипян В.О.* Разработка математических моделей систем защиты информации, содержащих диофантовы трудности: монография. – КубГУ, 2021.

УДК 004.491.42

И.А. Писарев, Л.К. Бабенко

Россия, г. Таганрог, Южный федеральный университет

ИСПОЛЬЗОВАНИЕ ЛОКАЛЬНЫХ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ ДЛЯ СКРЫТИЯ ТЕЛА ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В работе предлагается метод обхода антивирусной защиты, использующий вместо классического криптографического шифрования тела ВПО либо подгрузки тела с сервера злоумышленника, локальные большие языковые модели для генерации частей кода ВПО, которые средства антивирусной защиты опознают наиболее чувствительно, такие как, чтение и запись файлов, установка сетевого соединения с последующим обменом данных с каким-либо сервером. Разработан алгоритм и программный образец, позволяющий обойти защиту актуальной версий антивируса Microsoft Defender.

Ключевые слова: *большие языковые модели; вредоносное программное обеспечение; антивирусные средства.*

The work proposes a method for bypassing anti-virus protection that uses, instead of classical cryptographic encryption of the malware body or downloading the body from the attacker's server, local large language models to generate parts of the malware code that anti-virus protection tools recognize most sensitively, such as reading and writing files, installing a network connection followed by data exchange with any server. An algorithm and software sample have been developed that allows you to bypass the protection of the current versions of Microsoft Defender antivirus.

Keywords: *large language models; malware; antiviral agents.*

Введение

С развитием современных технологий нейросетей, в частности, больших языковых моделей, меняются многие правила существования человечества. На данный момент нейросети уже могут полностью заменить некоторые задачи, которые выполняют люди, а большинство задач становится намного легче делать при помощи нейросетей. С их помощью можно получить краткий пересказ статьи или

видео, получить развернутый ответ на свой вопрос при задании необходимого контекста, сделать анализ данных. С помощью больших языковых моделей также можно генерировать программный код по словесному описанию. Данная особенность позволяет заменить технику криптографического шифрования тела ВПО, поскольку в данном случае тело будет создаваться непосредственно нейросетевой моделью из своих предобученных весов, которые никак не могут быть опознаны антивирусными средствами защиты как вредоносные. Это позволит скрывать тело ВПО и эффективно обходить антивирусную защиту. Помимо этого, тело будет получаться локально без сетевых соединений, за которыми пристально наблюдают антивирусные средства.

Большие языковые модели

Одними из самых популярных больших языковых моделей являются ChatGPT, Claude, Gemini [1]. Данный модели доступны только через онлайн соединение. Энтузиасты исследователи нейросетей создали открытый репозиторий HuggingFace [2], куда команды разработчиков выкладывают свои предобученные модели, которые можно запускать локально на своем оборудовании. Поскольку для запуска моделей требуется большой объем оперативной памяти был проверен ряд легковесных моделей в задаче программирования. По результатам пробных запусков лучше всего показала себя модель “AIGym/deepseek-coder-1.3b-chat-and-function-calling” [3].

Метод подгрузки тела вредоносного программного обеспечения из локальной языковой модели

Метод будет рассматриваться на примере наиболее опасных ВПО, направленных на кражу данных (data stealer) [4]. Схема метода описана на рис. 1:

1. С помощью социальной инженерии достигается скачивание и открытие жертвой файла-контейнера. Это может быть документ Microsoft Office с макросами, установщик ПО формата .exe или .msi или любой другой контейнер определенного формата, подходящий для атаки через социальную инженерию в требуемом контексте.
2. Из файла-контейнера извлекается файл VBScript – запускатор.

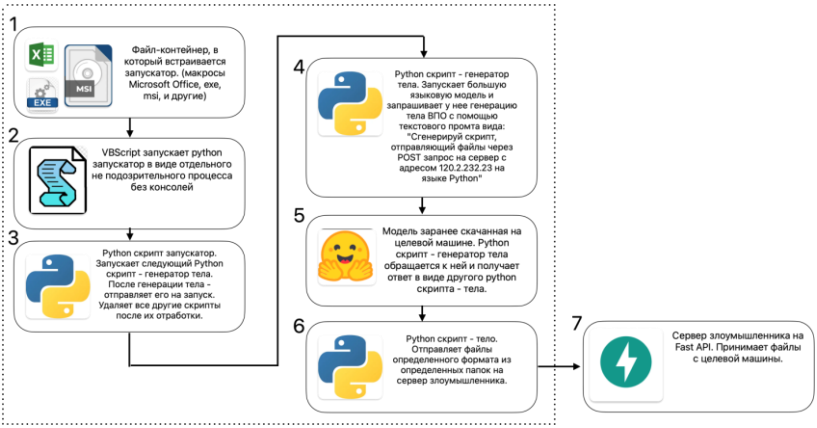


Рис. 1. Схема метода подгрузки тела из большой языковой модели

3. VBScript – запускатор запускает Python – запускатор.

4. Python – запускатор обращается к большой языковой модели. В случае если модель не выкачана на компьютере жертвы – осуществляется ее выкачивание с haggingface, после чего модель загружается в память и к ней осуществляется обращение в виде текстового запроса (промта) вида: "Сгенерируй скрипт, отправляющий файлы через POST запрос на сервер с адресом 120.2.232.23 на языке python"

5. После ответа большой языковой модели результат сохраняется в виде python – скрипта и отправляется на запуск.

6. Python – скрипт в данном случае выполняет вредоносные действия – осуществляет передачу файлов заданного формата из заданных папок на сервер злоумышленника.

7. Сервер злоумышленника принимает похищенные файлы. Может быть написан с помощью любого доступного фреймворка. В данном случае используется FastAPI.

Тестирование предложенного метода в лабораторных условиях на стенде из локальных виртуальных машин

Был создан тестовый стенд из виртуальных машин, где был осуществлен запуск программного образца. Используемая ОС – Windows 11 последней версии на момент написания данной статьи. Антивирус Windows Defender последней версии на момент написании данной статьи (рис. 2).

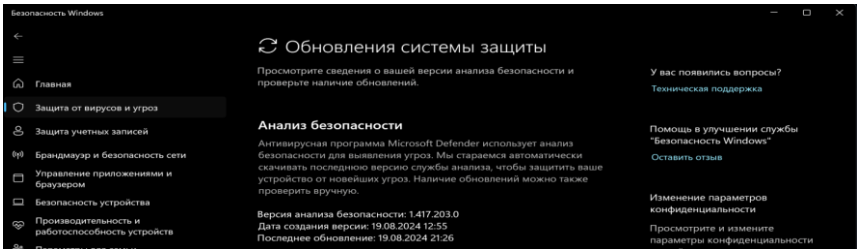


Рис. 2. Актуальная версия антивирусного средства Windows Defender

Стоит отметить, что для работы предлагаемого метода на данный момент требуется, чтобы на компьютере жертвы был установлен интерпретатор Python и библиотека transformers. В качестве контейнера использовался документ Microsoft Office с макросом. После открытия файла и включения макросов программный образец начал свою работу. Загрузка модели в память потребовала 5 ГБ оперативной памяти. Суммарный необходимый объем оперативной памяти для корректного запуска – не менее 9 ГБ (рис. 3).

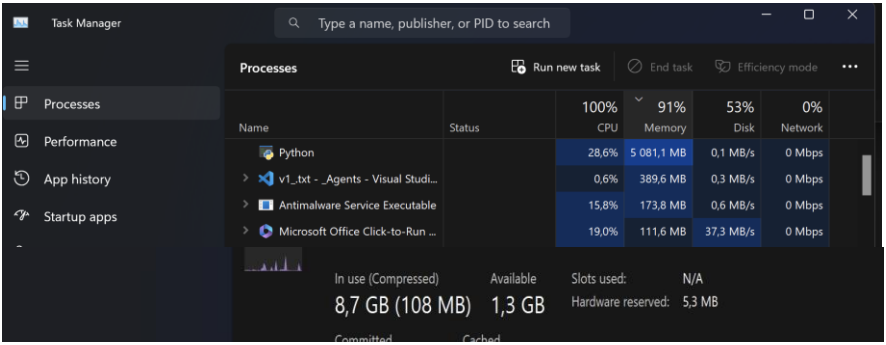


Рис. 3. Требуемый объем памяти

Время обращения к модели составило порядка 5 минут. Использовался виртуальный процессор с тактовой частотой 3.2 ГГЦ и 4 физическими ядрами (виртуальная машина на устройстве с процессором Apple Silicon M1 Pro). Стоит отметить, что главным требованием к компьютеру жертвы является только оперативная память.

Мощность процессора влияет на скорость ответа модели и на практике чем она будет больше, тем больше вероятность что антивирус пропустит данную атаку, поскольку время между критическими событиями (получение вредоносного файла – отправка данных на какой-то сервер) будет увеличено. В случае если бы после открытия вредоносного файла данные сразу бы передались на сервер злоумышленника это вероятнее всего привело бы к прерыванию процесса и удалению файла как вредоносного. По результату отработки на сервер пришел документ с компьютера жертвы в формате base64 (рис. 4).

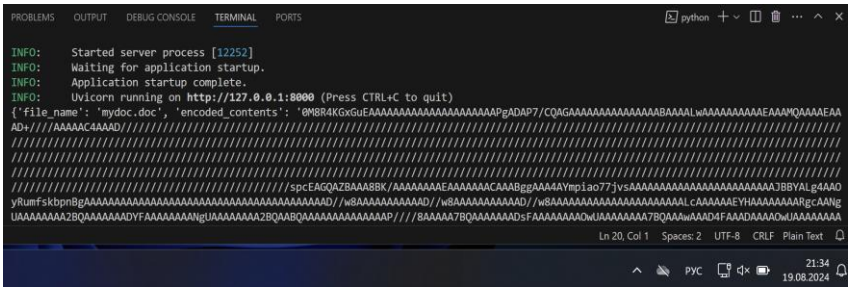


Рис. 4. Получение тестового файла на сервере злоумышленника

Работу вредоносного кода антивирусное средство не прерывало, после запуска вредоносный файл остался на диске в системе жертвы. Это подтверждает факт обхода антивирусного средства Microsoft Defender, которое обладает возможностью проведения сигнатурного анализа, анализа поведения, анализа в виртуальной среде (песочнице), а также мощного движка анализа вредоносных скриптов AMSI [5].

Заключение

В работе представлен метод обхода антивирусных средств на основе извлечения тела вредоносного программного обеспечения из локальной языковой модели. В дальнейшем планируется доработка метода, включающая в себя автоматическую установку всех необходимых зависимостей для запуска ВПО, а также тестирование на других антивирусных средствах.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Cartwright O., Dunbar H., Radcliffe T.* Evaluating privacy compliance in commercial large language models-chatgpt, claude, and gemini. – 2024.
2. *Jain S.M.* Hugging face // Introduction to transformers for NLP: With the hugging face library and models to solve problems. – Berkeley, CA: Apress, 2022. – P. 51-67.
3. HuggingFace: официальный сайт. – URL: https://huggingface.co/AIGym/deepseek-coder-1.3b-chat-and-function-calling (дата обращения: 10.08.2024).
4. *Kuraku S., Kalla D.* Emotet malware—a banking credentials stealer // Iosr J. Comput. Eng. – 2020. – Vol. 22. – P. 31-41.
5. *Hendler D., Kels S., Rubin A.* Amsi-based detection of malicious powershell code using contextual embeddings // Proceedings of the 15th ACM Asia Conference on Computer and Communications Security. – 2020. – P. 679-693.

УДК 004.63

К.С. Романенко, Е.А. Ищукова

Россия, г. Таганрог, Южный федеральный университет

АЛГОРИТМ ХРАНЕНИЯ ПРИВАТНЫХ ДАННЫХ В БЛОКЧЕЙН СИСТЕМАХ

В данной работе рассматривается проблема хранения приватных данных в блокчейн реестрах. Также представлен алгоритм хранения приватных данных непосредственно в блокчейн реестре. Данный алгоритм предполагает шифровать данные на клиенте и загружать данные в блокчейн в зашифрованном виде. В результате работы были рассмотрены достоинства и недостатки алгоритма, а также дальнейшие пути его развития.

***Ключевые слова**: Приватные данные; блокчейн; HyperLedger Fabric; конфиденг; Ethereum.*

This paper examines the problem of storing private data in blockchain registries. It also presents an algorithm for storing private data directly in the blockchain registry. This algorithm assumes encrypting data on the client and loading data into the blockchain in encrypted form. As a result of the work, the advantages and disadvantages of the algorithm, as well as further ways of its development, were considered.

***Keywords:** Private data; blockchain; HyperLedger Fabric; confidential; Ethereum.*

1. Введение

Блокчейн – это распределенная база данных, хранящая информацию в блоках, которые связываются друг с другом в цепочку. Каждый блок содержит запись о транзакциях, которые произошли в определенный момент, а вся цепочка защищена криптографией, что делает ее практически неизменяемой.

Представим, что каждый блок – это страница в большой книге, заполненная информацией о транзакциях. Эти страницы соединены между собой, чтобы сформировать непрерывную историю, которую невозможно изменить.

Перейдем к рассмотрению блокчейн платформ.

Блокчейн-платформа – это основа для создания и запуска децентрализованных приложений (dApps) и других сервисов, использующих технологию блокчейна. Она представляет собой набор инструментов и функций, позволяющих разработчикам создавать собственные приложения и решения, основанные на блокчейне.

Публичный блокчейн представляет собой распределенную базу данных, доступную для записи и чтения любому участнику сети. Его архитектура основана на децентрализованной модели, где нет единого центрального органа управления, а данные хранятся и синхронизируются между множеством узлов.

Bitcoin (BTC) стал первой и самой известной криптовалютой, основанной на публичном блокчейне. BTC предназначен для прямых переводов денег между людьми без посредников и банков.

Ethereum вышел за рамки простой криптовалюты и стал платформой для создания децентрализованных приложений (dApps). Ethereum имеет собственную виртуальную машину, которая позволяет выполнять умные контракты – автоматизированные программы, решающие разные задачи. Разработчики платформы Ethereum предложили стандарт ERC-20 для создания собственных токенов (криптовалют), что сделало его очень популярным.

Если публичные блокчейны – это открытые площади, то приватные – это закрытые клубы, доступные только для членов. Они не так публичны и не так безопасны, как публичные, но предоставляют большую гибкость и контроль.

Представьте себе, что вы работаете в большой компании и хотите использовать блокчейн для управления цепочками поставок. Вы можете использовать приватный блокчейн, где только ваши поставщики и ваши сотрудники смогут просматривать и изменять информацию о товарах.

Рассмотрим наиболее популярные приватные блокчейн платформы HyperLedger Fabric и Конфидент.

Hyperledger Fabric – это платформа с открытым исходным кодом, разработанная для создания частных и конфиденциальных блокчейнов, ориентированных на бизнес-приложения. Платформа позволяет настраивать сети под конкретные нужды, используя разные компоненты и модули. Это делает её очень гибким инструмен-

том для разработки различных решений. Hyperledger Fabric позволяет создавать “частные каналы”, где информация доступна только определенным участникам. Это обеспечивает конфиденциальность и безопасность данных. Платформа может масштабироваться для обработки больших объемов транзакций, что делает её пригодным для крупных корпораций и организаций.

Hyperledger Fabric является частью проекта Hyperledger, который разрабатывается фондом Linux Foundation.

Конфидент – это универсальная российская платформа блокчейна, которая позволяет создавать как открытые (публичные), так и закрытые (приватные) сети для широкого спектра задач в корпоративном и государственном секторах.

Конфидент предлагает гибкость в выборе криптографических алгоритмов, позволяя оптимизировать систему под конкретные требования проекта. В платформе реализованы два типа криптографии: Waves и ГОСТ.

Платформа Конфидент использует инфраструктуру открытых ключей (PKI) для обеспечения безопасности. PKI доступна только при использовании ГОСТ криптографии.

2. Проблема хранения приватных данных

Несмотря на то, что блокчейн известен своей прозрачностью и безопасностью, хранение приватной информации в нем представляет ряд проблем, которые требуют внимательного подхода.

В публичных блокчейнах все транзакции и данные публично доступны для просмотра всеми. Это делает их прозрачными, но также создает проблему конфиденциальности для чувствительной информации.

В частных сетях доступ к данным ограничен только участниками. Однако это создает риск централизации и непрозрачности.

Законы о конфиденциальности данных могут ограничивать хранение определенной информации в блокчейнах. Сложно определить ответственность в случае нарушения конфиденциальности данных в децентрализованных системах.

В статье [1] рассматривается новый протокол повышения конфиденциальности на основе смарт-контрактов Privacy Pools. Протокол вводит механизм, позволяющий пользователям раскрывать опре-

деленные свойства своей транзакции без необходимости раскрывать саму транзакцию. Основная концепция заключается в том, чтобы позволить пользователям публиковать доказательство с нулевым разглашением, демонстрирующее, что их средства (не) происходят из известных (не)законных источников, без публичного раскрытия всей истории своих транзакций.

В статье [2] рассматривается схема криптографического запутывания для смарт-контрактов на основе существующих механизмов блокчейна, стандартных криптографических предположений и свидетельского шифрования. В предлагаемой схеме запутанный смарт-контракт не раскрывает свой алгоритм и жестко закодированные секреты и сохраняет зашифрованные состояния. Любой пользователь может предоставить ему зашифрованные входные данные и разрешить не доверенной третьей стороне выполнить его.

В статье [3] представлен новый метод, позволяющий решить проблему приватности информации в открытых блокчейн системах с использованием криптографического протокола доказательства с нулевым разглашением zk-SNARK. Предложенный метод реализован в виде криптографической схемы на основе библиотеки libsnark. Для интеграции криптографической схемы в систему модифицирован Ethereum C++ клиент, куда добавлены новые функции и интерфейс для работы с ними в виде предкомпилированных контрактов.

На сегодняшний день существуют методы обеспечения конфиденциальности данных, основанные на протоколе с нулевым разглашением, а также реализации смарт-контрактов, адаптированных для конкретных задач с учётом разделения прав доступа.

3. Алгоритм обеспечения хранения приватной информации

Для решения поставленной задачи мы предлагаем реализовать смарт-контракт со следующим алгоритмом работы. Рассмотрим двух пользователей, относящихся к первой категории пользователей. Пользователь А обрабатывает и помещает данные в блокчейн, пользователь В хочет и имеет право получить эти персональные данные для обработки. Для пользователей А и В сформированы пары ключей в системе, соответственно (PubKa, PrivKa) и (PubKb, PrivKb) Алгоритм взаимодействия пользователей А и В сводится к следующему (рис. 1).

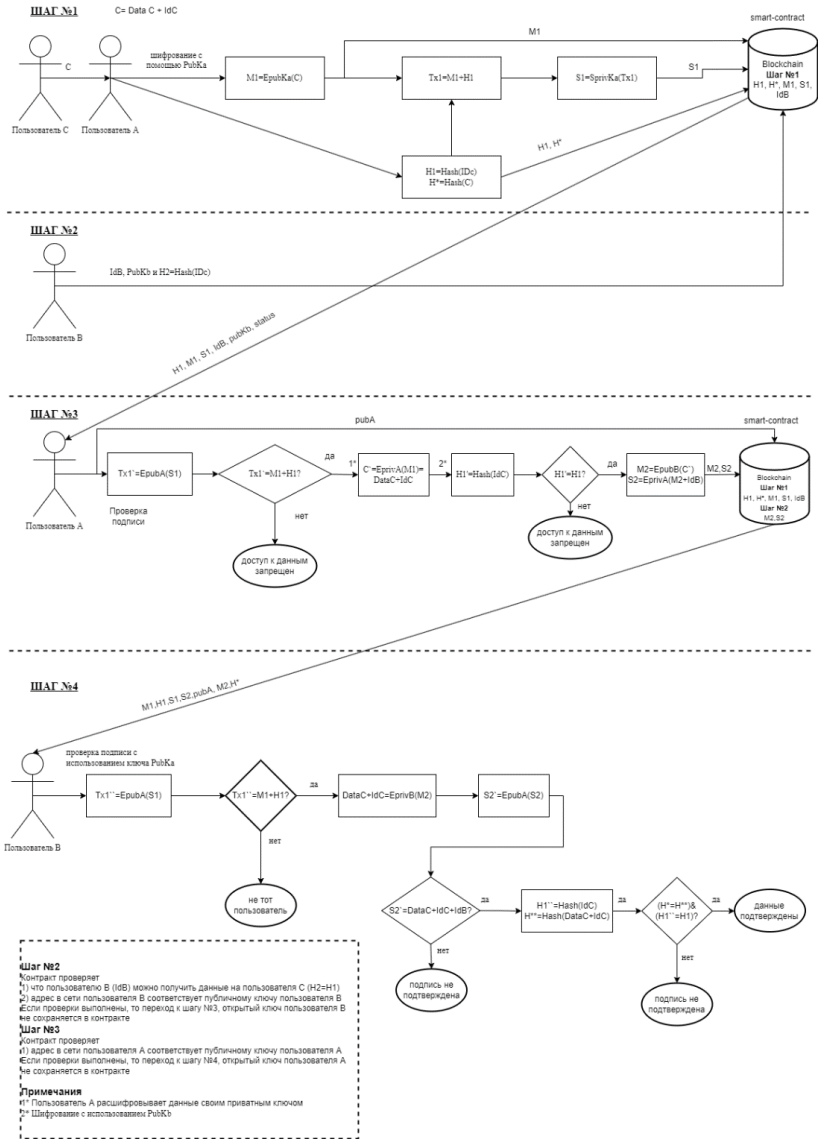


Рис. 1. Алгоритм хранения приватных данных в блокчейн

Пользователь А шифрует данные (Data C) пользователя С и идентификатор пользователя С с использованием своего открытого ключа, полученные зашифрованные данные (данные пользователя С + идентификатор пользователя С) записывает в M1. Далее пользователь А формирует транзакцию Tx1, которая состоит из зашифрованных данных M1 и хэша H1 взятого от значения IdC. Затем пользователь А подписывает транзакцию Tx1 с помощью своего приватного ключа, формируя подпись S1. Также пользователь А формирует хэш H* от данных пользователя Data C и идентификатора пользователя IdC (C=Data C + IdC). На этом шаге пользователь А заносит в блокчейн значения H1, H*, M1, S1. При этом в контракте устанавливаются флаги в зависимости от того, каким пользователям (по их идентификаторам) могут быть доступны занесенные данные (в рассматриваемом примере данные доступные пользователю В).

Пользователь В делает запрос в контракт на получение данных пользователя С. Для этого он в контракт передает данные, которые состоят из идентификатора пользователя В (IdB), публичного ключа пользователя В (PubKb, не хранится в блокчейне, а передается пользователю А) и хэша идентификатора пользователя С (H2=Hash(IdC)). Контракт проверяет, что в блокчейне есть данные пользователя С (H2=H1), а также проверяет, что пользователю В (IdB) можно получить данные на пользователя С. Также контракт проверяет, что адрес в сети пользователя В соответствует публичному ключу пользователя В (для получения публичного адреса аккаунта берутся последние 20 байтов хэша Keccak-256 от публичного ключа и добавляется 0x в начало). Если проверки выполнены положительно, то контракт отправляет запрос к пользователю, который поместил в контракт данные пользователя С (т.е. к пользователю А). При этом контракт передает пользователю А открытый ключ пользователя В и флаг status (он содержит информацию о том, что разрешен доступ к данным).

Пользователь А проверяет данные, полученные от контракта с помощью своего публичного ключа. Для этого он восстанавливает значение Tx1` (Tx1`=EpubA(S1)) и сравнивает его с данными, полученными от контракта (Tx1`=M1+H1?). Если проверка пройдена положительно, то расшифровывает своим приватным ключом данные (C`=EprivA(M1)=DataC+IdC) и проверяет хэш идентификатора, чтобы убедиться в том, что данные принадлежат пользователю

C (H1`=Hash(IdC) и H1`=H1?). Далее пользователь A готовит данные для пользователя B. Для этого он шифрует данные публичным ключом пользователя B (M2=EpubB(C`)) и подписывает данную транзакцию своим приватным ключом (S2=EprivA(M2+IdB). Пользователь A отправляет в контракт данные M2, S2, pubA (публичный ключ пользователя A не сохраняется в блокчейн). Контракт проверяет, что адрес в сети пользователя A соответствует публичному ключу пользователя A (Для получения публичного адреса своего аккаунта берутся последние 20 байтов хэша Keccak-256 от публичного ключа и добавляется 0x в начало.). Если проверки выполнены, то переход к шагу №4, открытый ключ пользователя A не сохраняется в контракте и в случае успешной проверки передает пользователю B данные (M1, H1, S1, S2, pubA, M2, H*).

Пользователь B проверяет публичным ключом пользователя A подпись данных (Tx1``=EpubA(S1) и Tx1``=M1+H1?), и затем расшифровывает данные своим приватным ключом (DataC+IdC=EprivB(M2)). Далее пользователь проверяет подпись (S2`=EpubA(S2) и S2`=DataC+IdC+IdB?) и проверяет, что данные действительно из реестра блокчейн (H1``=Hash(IdC) и H**=Hash(DataC+IdC). Если условие проверки хэшей выполняется «(H*=H**)&(H1``=H1)», то можно говорить о том, что данные, которые лежат в блокчейне действительно принадлежат пользователю C, пользователь B получил эти данные и может их использовать.

Разработанный алгоритм позволяет хранить приватные данные непосредственно в блокчейн реестре и в результате нет необходимости разворачивать какую-либо систему управления базами данных непосредственно на клиентском устройстве.

Заключение

В результате данной работы была рассмотрена проблема хранения приватных данных как в публичных, так и в приватных блокчейн платформах. Также был представлен алгоритм, позволяющий решить проблему хранения приватных данных как в публичных, так и в приватных блокчейн платформах. В дальнейшем планируется разработать оптимизированную версию алгоритма, в которой большинство проверок данных и ключей будет выполнятся на стороне клиента и тем самым снизить нагрузку непосредственно на блокчейн реестр.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Vitalik Buterin, Jacob Illum, Matthias Nadler, Fabian Schär, Ameen Soleimani*. Blockchain privacy and regulatory compliance: Towards a practical equilibrium // Blockchain: Research and Applications. – 2024. – Vol. 5, Issue 1. – 100176. – ISSN 2096-7209. – https://doi.org/10.1016/j.bcra.2023.100176.
2. *Sora Suegami.* Cryptographic obfuscation for smart contracts: Trustless bitcoin bridge and more [Текст] // Blockchain: Research and Applications. – 2023. – No. 4. – https://doi.org/10.1016/j.bcra.2022.100118.
3. *Кондырев Д.О.* Разработка метода сокрытия приватных данных для системы тендеров на основе технологии блокчейн // Прикладная дискретная математика. – 2020. – № 48. – С. 63-81. – DOI: 10.17223/20710410/48/6. – EDN CHKZFB.

УДК 004.93

К.Е. Румянцев[1], Л.К. Хаджиева[2]

[1]Россия, г. Таганрог, Южный федеральный университет

[2]Россия, г. Грозный, Грозненский государственный нефтяной технический университет им. М.Д. Миллионщикова

АНАЛИЗ ПРИМЕНЕНИЯ НЕЙРОЛИНГВИСТИЧЕСКОЙ ИДЕНТИФИКАЦИИ СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Проведен анализ результатов исследований нейролингвистической текстовой идентификации систем искусственного интеллекта. Целью исследования являлось проведения анализа результатов нейролингвистической текстовой идентификации систем искусственного интеллекта. Результаты анализа показывают возможность применения нейролингвистической текстовой идентификации систем искусственного интеллекта в качестве основных параметров. В предыдущем исследовании авторов приведены зависимости основных и производных параметров нейролингвистической текстовой идентификации от изменений систем искусственного интеллекта, а также проанализированы изменения показателей систем искусственного интеллекта по основным (информационная емкость, энтропия, избыточность) и производным (коэффициент избыточности, коэффициент вербальности) параметрам. В результате анализа установлено, что значения основных и производных параметров нейролингвистической текстовой идентификации систем искусственного интеллекта при постановке одной и той же задачи разным систем искусственного интеллекта значительно отличаются.

Ключевые слова: *анализ; исследование; искусственный интеллект; нейролингвистическая идентификация; избыточность; информационная.*

The analysis of the results of research on the neuro-linguistic textual identification of artificial intelligence systems is carried out. The purpose of the study was to analyze the results of neuro-linguistic textual identification of artificial intelligence systems. The results of the analysis show the possibility of using neuro-linguistic textual identification of artificial intelligence systems as the main parameters. In the previous study of the authors, the dependences of the main and derived parameters of neuro-linguistic textual identification on changes in artifi-

cial intelligence systems are presented, and changes in the indicators of artificial intelligence systems by basic (information capacity, entropy, redundancy) and derivative (redundancy coefficient, verbality coefficient) parameters are analyzed. As a result of the analysis, it was found that the values of the basic and derived parameters of the neuro-linguistic textual identification of artificial intelligence systems when setting the same task differ significantly from different artificial intelligence systems.

Keywords: *artificial intelligence; neuro-linguistic identification; redundancy; information capacity.*

Введение

Нейролингвистическая идентификация представляет собой идентификацию интеллектуальных систем по параметрам идентификатора, отражающего мозговые механизмы речевой деятельности.

Результатом речевой деятельности может выступать текст, что объясняет существование нейролингвистической текстовой идентификации.

Установлено что фактором идентификации может являться язык речевой деятельности. В качестве параметров идентификатора в данном случае выступают: коэффициент избыточности, коэффициент вербальности, эмпирическая энтропия, информационная емкость.

В статье решается проблема применения разработанного подхода к идентификации систем искусственного интеллекта.

Постановка задачи

Актуальна задача оценки возможности применения разработанного подхода для нейролингвистической текстовой идентификации систем искусственного интеллекта. Успешное решение этой задачи дает возможность показать индивидуальность систем искусственного интеллекта.

Целью является анализ результатов исследования нейролингвистической текстовой идентификации систем искусственного интеллекта и определение вероятности использования параметров в качестве факторов нейролингвистической идентификации систем искусственного интеллекта [1–3].

Описание исследований

В исследовании проведен анализ ряда систем искусственного интеллекта: Gerwin, YandexGPT, RoboGPT, CopyMonkey, Micrasoft Copilot, Всезнайка, ChatGPT 3.5. Всем исследуемым системам ИИ определена одна и та же задача для решения. Постановка задачи формулируется на русском языке. Первоначально предполагается, что при постановке одной и той же задачи на одном языке разные системы ИИ должны выдавать одинаковый результат. Однако, исследование показывает, что результат каждой системы ИИ индивидуален и отличается от всех остальных [4–6].

Результаты исследований

Проанализированы параметры, полученные в некоторых системах искусственного интеллекта разных поколений. Исследованные семь систем искусственного интеллекта относятся к разным поколениям и являются как российскими, так и иностранными.

Результаты оценки основных параметров нейролингвистической текстовой идентификации систем искусственного интеллекта сведены в табл. 1.

Таблица 1

Параметры нейролингвистической текстовой идентификации систем искусственного интеллекта

Системы искусственного интеллекта	Информационная емкость H_{Bmax}	Энтропия H_B	Избыточность B
Gerwin	5,42	1,73	3,69
YandexGPT	5,93	1,79	4,14
Micrasoft Copilot	6,02	1,85	4,17
RoboGPT	5,24	1,70	3,53
CopyMonkey	5,42	1,74	3,67
Всезнайка	5,67	1,93	3,74
ChatGPT 3.5	3,80	1,65	2,14

Результаты расчётов производных параметров нейролингвистической текстовой идентификации систем искусственного интеллекта: коэффициента избыточности и коэффициента вербальности сведены в табл. 2

Таблица 2

Производные параметры систем искусственного интеллекта

Системы искусственного интеллекта	Коэффициент избыточности, μ_B	Коэффициент вербальности, G_B
Gerwin	0,680	0,470
YandexGPT	0,698	0,432
Micrasoft Copilot	0,692	0,443
RoboGPT	0,674	0,482
CopyMonkey	0,678	0,474
Всезнайка	0,659	0,516
ChatGPT 3.5	0,561	0,772

В качестве параметров идентификатора в данном случае выступают коэффициент избыточности, коэффициент вербальности, эмпирическая энтропия, информационная емкость. Все поставленные изначально задачи решены. Определено, что значения основных и производных параметров нейролингвистической текстовой идентификации систем искусственного интеллекта при постановке одной и той же задачи разным системам ИИ значительно отличаются [7, 8]. Проанализированы основные и производные параметры нейролингвистической текстовой идентификации систем искусственного интеллекта. Определены средние значения основных и производных параметров нейролингвистических текстовых идентификаторов систем искусственного интеллекта.

Заключение

Проведен анализ результатов исследований семи систем искусственного интеллекта разных поколений. Системам ИИ определена одна и та же задача на русском языке для решения. Анализ результатов показывает, что результат каждой системы ИИ индивидуален и отличается от всех остальных. Таким образом наблюдается, что показатели параметров систем ИИ третьего поколения значительно отличаются от показателей параметров систем ИИ четвертого поколения. При переходе от одной системы ИИ к другой наблюдается изменение параметров.

В результате проведенных исследований установлено, что значения основных и производных параметров нейролингвистической текстовой идентификации систем искусственного интеллекта при постановке одной и той же задачи разным СИИ значительно отличаются.

Исследования открывают возможность новых подходов к обнаружению идентификации и аутентификации систем искусственного интеллекта. Полученные результаты позволяют показать индивидуальность систем искусственного интеллекта откуда следует возможность существования в них задатков моделирования мыслительной деятельности.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Котенко В.В.* Технологии информационного анализа пользовательского уровня телекоммуникационных систем: учебное пособие. – Ростов-на-Дону – Таганрог: Изд-во ЮФУ, 2019. – 194 с.
2. *Котенко В.В.* Теория виртуализации и защита телекоммуникаций: монография. – Таганрог: Изд-во ТТИ ЮФУ, 2011. – 236 с.
3. *Котенко В.В. Румянцев К.Е., Котенко С.В.* Методология идентификационного анализа инфокоммуникационных систем: монография. – Ростов-на-Дону: Изд-во ЮФУ, 2014. – 315 с.
4. *Хаджиева Л.К., Котенко В.В., Румянцев К.Е.* Нейролингвистическая информационная идентификация интеллектуальных систем // Известия ЮФУ. Технические науки. – 2024. – № 3. – С. 33-44. – ISSN 1999-9429. – DOI: 10.18522/2311-3103-2024-3-33-44.
5. *Juan Liu, Min Hu, Ying Wang, Zhong Huang, Julang Jiang*. Symmetric Multi-Scale Residual Network Ensemble with Weighted Evidence Fusion Strategy for Facial Expression Recognition // Symmetry. – 2023. – 15 (6). – P. 1228. – https://doi.org/10.3390/sym15061228.
6. *Ямченко Ю.В.* Методы решения задач аутентификации и идентификации пользователя на основе анализа клавиатурного почерка // Вестник Московского государственного технического университета им. Н.Э. Баумана. Серия Приборостроение. – 2020. – № 1 (130). – С. 124-139.
7. *Куртукова А.В.* Разработка интеллектуальной системы для идентификации автора исходного кода на основе нейронных сетей. – Текст: непосредственный // Молодой ученый. – 2020. – № 40 (330). – С. 2-3.
8. *Куртукова А.В., Романов А.С.* Проблема искусственно сгенерированных исходных кодов в задаче идентификации автора программы // Сборник избранных статей научной сессии ТУСУР. – 2022. – № 1-2. – С. 131-134.

УДК 004.942

А.А. Рыженко

Россия, г. Москва, Финансовый университет

ЗОНА НЕДОВЕРИЯ БИБЛИОТЕК QR CODE

Аннотация: в 2015 году вступил в силу ГОСТ ИСО МЭК 18004-2015 [1], фактически являющийся дубль переводом ISO/IEC 18004-2015 [2]. ГОСТ имеет собственную более чем 10-летнюю историю роста и развития, что в положительном аспекте отразилось на содержимом приложений и описанных алгоритмов. Статус документа как рекомендательного использован с точки зрения внедрения новшеств при реализации гигантами компьютерной индустрии в разрез с положениями ГОСТ (что в последствии стало коммерческой тайной). Здесь необходимо отметить, что и в самом стандарте есть ряд алгебраических ошибок. Данные моменты также будут отражены далее.

Ключевые слова: *QR Code; quick response code; алгоритмы; информационная безопасность.*

In 2015, GOST ISO IEC 18004-2015 [1] came into force, which is actually a duplicate translation of ISO/IEC 18004-2015 [2]. GOST has its own more than 10-year history of growth and development, which has had a positive effect on the content of the applications and described algorithms. The document's status as a recommendation was used from the point of view of introducing innovations in the implementation of computer industry giants in violation of the provisions of GOST (which subsequently became a commercial secret). It should be noted here that the standard itself contains a number of algebraic errors. These points will also be reflected below.

Keywords: *QR Code; quick response code; algorithms; information security.*

Введение

Современные алгоритмы формирования бинарной матрицы дискретного поля используют модели и последовательности, разработанные десятилетиями назад. Перевод одной системы счисления в другую также хорошо известен в сфере деятельности, связанной с оцифровкой данных произвольного формата. Существующая система хранения данных в большей мере использует именно двоичную форму представления. Для трансляции и тиражирования информации также при-

меняют двоичный формат для исходных и результирующих данных. Как следствие, применение двоичной матрицы для графического представления кода быстрого отклика (*quick response code*, *QR Code*) является естественным шагом очередной формы представления данных. Стоит также упомянуть, что форма *дизеринга* (*dither*) печатных изданий использует дискретные пространства для отображения изображений любого формата с использованием исключительно базовых цветовых каналов [3]. И третья важная составляющая – существующие модели контроля сумм данных (например, алгоритм Рида-Соломона), позволяющие находить и исправлять ошибки и стирания автоматически, используются не только при передаче данных в зашумленных каналах связи, но и в теории автоматов при программировании триггеров. В результате, синтез методики формирования дискретного поля, теории рекурсивного представления алгебраических фракталов, основ классификации качественных показателей мягкой логики (*fuzzy logic*), алгоритмов корректирующего кода контроля сумм и пр. позволили разработать такое направление, как графическое представление кода быстрого отклика или *QR Code*. Разновидностью данного направления можно вполне считать *Data Matrix* (упрощенный вариант) [4], *Aztec Code* (усложненный вариант) [5] и др.

В качестве упрощения дальнейшего анализа представленного материала будет использован самый минимальный уровень *QR Code – Micro QR Code ver. M1*. Размерность поля и матрицы данных всего 11×11.

Согласно ГОСТ, на данном уровне можно закодировать только числовую последовательность длинной не более 5 символов и с отсутствием возможности добавления корректирующего кода даже для исправления одной ошибки. Другими словами, согласно теории мягкой логики в данном случае должен быть уровень *N* (*nothing* или *NULL*). Но уровень не объявлен в стандарте и забыт на последующих версиях кода. На данную странность в статье далее также будет представлено возможное решение.

Для простоты отображения полученных результатов использован табличный процессор автоматизированного офиса со встроенным математическим аппаратом функций. В качестве примера реализации полного алгоритма кодирования и декодирования можно использовать источники [6]. Дополнительных формул для операций *XOR*, а также для детального поиска в полях Галуа (прямой и обратный поиск) в текущей версии офиса не потребовалось.

Исходные данные: пять одинаковых цифр – 77777. Согласно ГОСТ разбиваем на числа в три и два разряда, и переводим в двоичный формат. Далее добавляем служебную информацию. В текущей версии *M1* не потребовалось дополнять нулевой хвост в конце исходной последовательности (первая странность стандарта). Но, в ГОСТ упомянуто, что если полученная последовательность не будет кратна 8, то необходимо добавить 4 нуля. И здесь есть вторая странность – по алгоритму Хэмминга анализ последовательности осуществляется справа налево (аналогично используется в коде при декодировании *QR Code*), т.е. нули должны быть перед исходным кодом.

В примере третье число 1101. По алгоритму Хэмминга должны были получить последовательность 0000 1101. Но, так как в ГОСТ это не отражено, то в данном примере и в последующем используется предложение *GAFAM*, т.е. нули добавляются после исходного кода, получаем 1101 0000. А в десятичном представлении вместо числа 13 получаем 208.

Алгоритм Рида-Соломона для подготовки последовательности *CRC*-кода широко представлен во многих публикациях как в форме алгебраического представления, так и в виде алгоритмов [7]. Но в самом ГОСТ не упоминается рекомендуемая вариация алгоритма, просто представлен пример и возможный вариант реализации в форме программного кода на языке программирования. Этим обстоятельством воспользовалась ассоциация *GAFAM* и внедрила в свои библиотеки *Segno*[1] на *PHP* или *qrcode.min.js*[2] на *JavaScrip* и др. собственный алгоритм реализации в разрез с классическими теории автоматов [10–12]. На текущий момент все мобильные и стационарные устройства оснащенные операционными системами ассоциации *GAFAM* используют исключительно данную библиотеку с всевозможными надстройками [13–15]. Особенностей у данного алгоритма множество. Например, последовательность шагов *XOR* анализа данных на поле Галуа алгоритма по своей структуре больше похож на известный алгоритм кодирования *AES* [8]. В статье детализация этапов кодирования и декодирования не предусмотрена, но представлена на известном Интернет-портале в виде последовательности дейст-

[1] *https://segno.readthedocs.io/en/latest/.*
[2] *https://cdnjs.cloudflare.com/ajax/libs/qrcodejs/1.0.0/qrcode.min.js.*

вия [6]. В итоге получили два числа корректирующего кода – 50 и 22. Добавляем данные числа в исходную последовательность и переводим в двоичный формат.

На следующем шаге необходимо перенести полученную последовательность на поле дискретной матрицы справа налево с нижнего угла и подвергнуть маскированию фрактальными поверхностями для выбора рационального варианта. В ГОСТ для каждой версии кода представлен собственный маршрут обхода (рис. 1). Уголком указано начало отсчета для каждого блока фиксированной длины. В более старших версиях кода блоки представлены не только прямоугольным форматом, но и в искаженной форме со смещением.

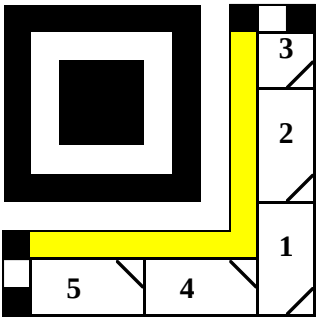


Рис. 1. Маршрут заполнения последовательностью бит матрицы по ГОСТ

Как можно заметить, большая часть поля отводится на заполнение именно корректирующим кодом. Данная особенность вполне оправдывает при поиске ошибок последовательности полученного кода в сильно зашумленных каналах связи и при контроле суммы больших архивов данных известных форматов (*ZIP*, *RAR* и пр.). Но насколько данный выбор алгоритма корректирующего кода обоснован для дискретной матрицы и почему заполнение идет не малыми блоками для «*размазывания*» исходного кода, а последовательно одним блоком останется загадкой ГОСТ.

Предпоследний этап кодирования полученной последовательности данных является выбор маски. Для микрокода в ГОСТ предлагается использовать четыре маски (для полного кода – 8 масок). В стандарте предлагается только 4 функции получения алгебраиче-

ского фрактала, но не упоминается, что данные варианты являются *XOR* по отношению друг к другу [4]. С другой стороны, данный фактор упущен намеренно, так как не все предложенные варианты являются взаимоисключающими. Именно этот момент далее приводит к существенной ошибке, которую *GAFAM* решила по-своему в разрез со стандартом.

Воспользуемся предлагаемыми стандартом шаблонами маскирования. Полученные вариации представлены на рис. 2. Забегая вперед, даже если просто обзорно посмотреть на результат можно заметить, что функция *XOR* отсутствует для перекрестных версий.

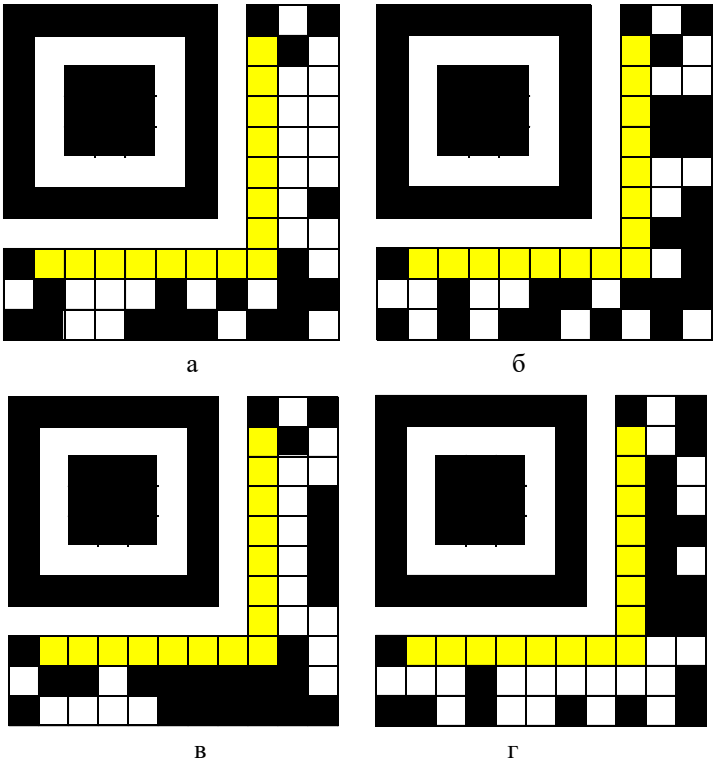


Рис. 2. Варианты маскирования 77777 вер. M1 по ГОСТ

На рис. 3 отображена та самая серьезная ошибка, описанная выше. Три варианта из четырех возможных привели к одинаковому результату. Данная коллизия никак не рассматривается в ГОСТ. Возможны два решения: второй вариант верный – по классической последовательности чтения информации или четвертый вариант верен – по алгоритму Хэмминга [9]. Программисты *GAFAM* пошли другим путем (рассмотрен ниже), но на собственном алгоритме выбирают классический первый вариант при использовании любых библиотек.

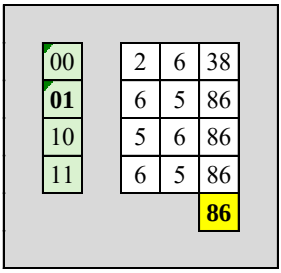


Рис. 3. Коллизии при нахождении максимума из максимумов по ГОСТ

Для данного примера выбираем первый верный вариант (рис. 4). Полученный результат невозможно проверить программными приложениями ассоциации ни в сети Интернет, ни с помощью мобильных устройств, так как реализация программных библиотек идет в разрез с ГОСТ.

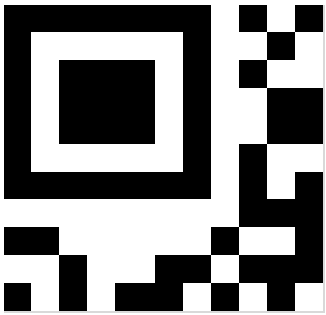


Рис. 4. Micro QR Code, M1, Numeric – 77777, ISO, binary (no errors)

Далее рассмотрим вариант реализации *M1*, предложенный ассоциацией. Несколько моментов, связанных с различием с ГОСТ, уже описаны выше. На рис. 5 представлен другой маршрут заполнения дискретного поля, разработанный ассоциацией. Как можно увидеть, нижняя часть поля строится абсолютно по-другому.

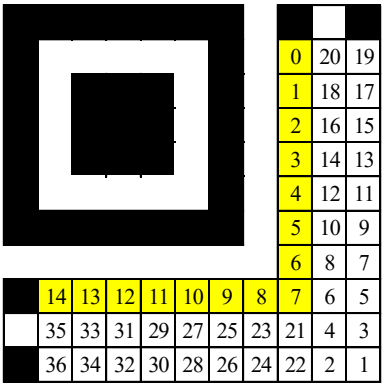
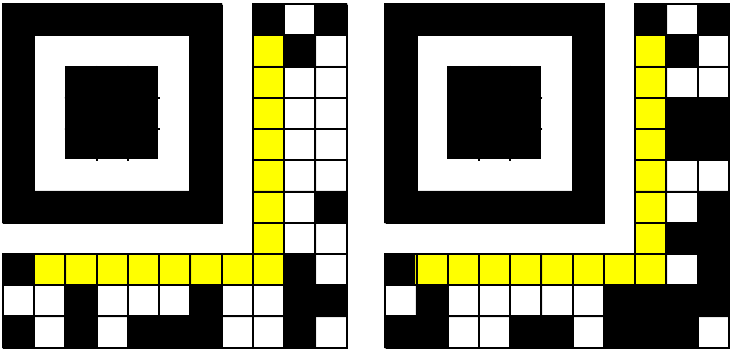
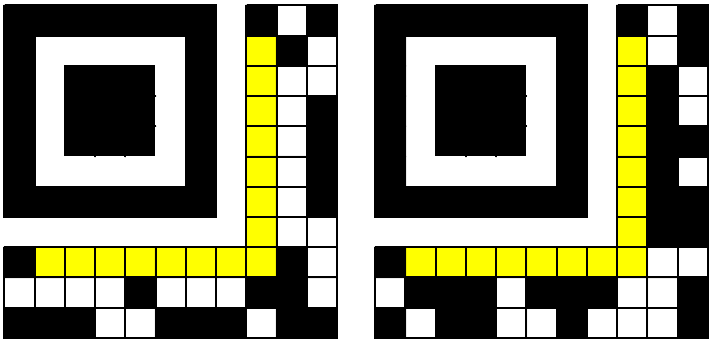


Рис. 5. Маршрут заполнения последовательностью бит матрицы не по ГОСТ

Четыре варианта после *XOR* маскирования исходной битовой последовательности не по ГОСТ представлены на рис. 6.



а б



в г

Рис. 6. Варианты маскирования 77777 вер. M1 не по ГОСТ

Можно действительно отметить, что небольшое изменение в маршруте обхода способствовало устранению коллизии по ГОСТ. На рис. 7 отображен полученный результат. Наибольший вариант представлен только в одном случае и выбор результирующей маски очевиден.

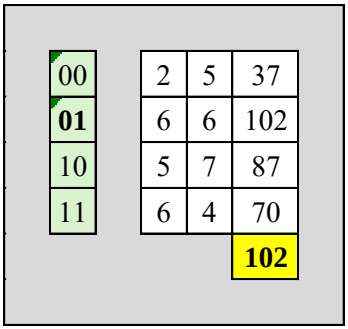


Рис. 7. Отсутствие коллизий при нахождении максимума из максимумов не по ГОСТ

Пример полученного результата не по ГОСТ представлен на рис. 8. Данный вариант можно прочитать любым доступным сканером *Micro QR Code*.

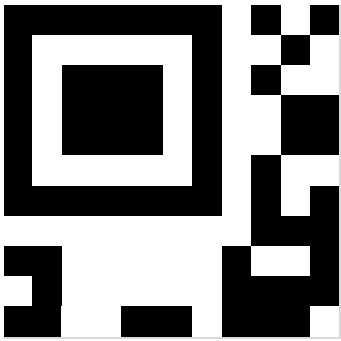


Рис. 8. Micro QR Code, M1, Numeric – 77777, по ISO, binary (no errors)

Дальнейший анализ данного варианта обхода показал, что коллизии возникают, хотя и с меньшей вероятностью. Следует также отметить, что проблему избыточности служебной информации на дискретном поле ассоциация не устранила.

Помимо представленных в статье моментов отклонений от ГОСТ алгоритмов кодирования и декодирования информации в библиотеках ассоциации, аналогичных изменений достаточно много. Некоторые уже представлены на открытом портале сети Интернет [6]. Другие уже расшифрованы, но не доведены до общественности. Возможно, программисты действительно предприняли попытку исправить некоторые алгебраические ошибки стандарта, но почему тогда новые достижения не доступны. Также необходимо учесть, что библиотека сырая и часто выдает ошибки из-за внутренних конфликтов. Например, на всех порталах можно создать *QR Code* версий *M* и *L*. С остальными уровнями тяжелее, далеко не все информационные ресурсы предоставляют такую возможность.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ ИСО МЭК 18004-2015. – Режим доступа: https://standartgost.ru/ГОСТ_Р_ ИСО/МЭК_18004-2015.
2. ISO/IEC 18004-2015. – Режим доступа: https://gcore.jsdelivr.net/gh/tonycrane/ tonycrane.github.io/p/ 409d352d/ISO_IEC18004-2015.pdf.
3. Дизеринг: зашумляем сигнал... – Режим доступа: https://habr.com/ru/articles/481386/.
4. Data Matrix. – Режим доступа: https://ru.wikipedia.org/wiki/Data_Matrix.

5. Aztec Code. – Режим доступа: https://ru.wikipedia.org/wiki/Aztec_Code.
6. Micro QR Code… – Режим доступа: https://habr.com/ru/users/Litloc/publications/articles/.
7. Код Рида-Соломона. – Режим доступа: https://habr.com/ru/articles/518120/.
8. AES – американский стандарт шифрования. – Режим доступа: https://habr.com/ru/ articles/508442/.
9. Корректирующие коды. Начало новой теории кодирования. – Режим доступа: https://habr.com/ru/ articles/511348/.
10. Отслеживание QR-кода (C# и C++). – Режим доступа: https://learn.microsoft.com/ru-ru/windows/mixed-reality/develop/native/qr-code-tracking-cs-cpp.
11. Общие сведения об отслеживании QR-кодов. – Режим доступа: https://learn. microsoft.com/ru-ru/windows/mixed-reality/develop/advanced-concepts/qr-code-tracking-overview.
12. QR-коды в Unity. – Режим доступа: https://learn.microsoft.com/ru-ru/windows/mixed-reality/develop/unity/qr-code-tracking-unity.
13. Системы координат. – Режим доступа: https://learn.microsoft.com/ru-ru/windows/mixed-reality/design/coordinate-systems.
14. Описание службы "Пространственные привязки Azure". – Режим доступа: https://learn.microsoft.com/ru-ru/azure/spatial-anchors/overview.
15. Запустите пример приложения: Android – Android Studio (Java или C++/NDK). – Режим доступа: https://learn.microsoft.com/ru-ru/azure/spatial-anchors/quickstarts/get-started-android?tabs=azure-portal%2Copenproject-java.

УДК 004.056

С.В. Селигеев, В.Г. Жуков

НАЦИОНАЛЬНАЯ СИСТЕМА ИМЕНОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В УПРАВЛЕНИИ УЯЗВИМОСТЯМИ

В работе рассматриваются теоретические и практические аспекты систем именования программного обеспечения, а также существующие проблемы их применения в процессах управления уязвимостями. На основании анализа проблем авторами предлагается концепция создания национальной системы именования программного обеспечения в рамках существующего банка данных угроз ФСТЭК.

Ключевые слова: *управление уязвимостями; управление активами; CPE.*

The paper considers theoretical and practical aspects of software naming systems, as well as existing problems of their application in vulnerability management processes. Based on the analysis of the problems, the authors propose the concept of creating a national software naming system within the existing FSTEC threat database.

Keywords: *vulnerability management; asset management; CPE.*

Введение

В рамках организации работ по управлению уязвимостями необходимо в первую очередь решить вопросы, связанные с идентификацией и классификацией активов по степени их влияния на бизнес-процессы [1, 2]. Наличие актуальной инвентаризационной информации, в первую очередь, об операционной системе и используемом программном обеспечении в привязке к сетевым идентификаторам (IP, FQDN) позволяет обнаруживать уязвимости, основываясь лишь на данных из системы управления активами, без непосредственного сетевого взаимодействия с активами [3]. Однако, это возможно только при одном условии – существование и использование одинакового уникального идентификатора программного обеспечения, как в описании уязвимости, так и описании актива. Все производители специализированных решений по управлению уязвимостями вынуждены использовать такие системы однозначного именования активов,

среди которых стандартом де-факто является Common Platform Enumeration (CPE) [4], как неотъемлемая часть Security Content Automation Protocol (SCAP) [5]. CPE определяет правила поименования программного обеспечения и пакетов, которые используются в Common Vulnerabilities and Exposures (CVE), для идентификации программного обеспечения, в котором потверждено наличие уязвимости. Запись CPE формируется в момент регистрации CVE уязвимости в National Vulnerability Database (NVD), и, в последствии, попадает в словарь (справочник) CPE, который доступен как виде файла (периодически обновляемого), так и посредством API.

NVD, как и другие агрегаторы данных об уязвимостях из недружественных стран, например, CISA Known Exploited Vulnerabilities (KEV) Catalog в связи с геополитической обстановкой могут в любой момент прекратить предоставлять услуги на территории Российской Федерации. Данная проблема является основной, но не единственной предпосылкой к созданию национальной системы именования программного обеспечения, с целью обеспечения независимости от внешних поставщиков информации, актуальность данного вопроса так же подтверждается рядом публикаций [6, 7]. Второй и не менее важной предпосылкой являются существующие проблемы практического применения CPE в процессе управления уязвимостями. Так, каждая запись в NVD содержит в себе, помимо описания уязвимости и оценки ее опасности, CPE-имя, которое отражает подверженное уязвимости конкретное программное обеспечения с указанием его версии. При наличии в системе управления активами описания программного обеспечения в виде CPE-имени открывается возможность обнаружения активов с уязвимостями через использование только лишь данных из системы управления активами при максимальной скорости и минимальных ресурсах.

В рамках проводимого исследования в качестве системы управления активами использовался GLPI с плагином Fields, для возможности добавления дополнительных информационных полей – CPE-имя. В качестве специализированного инструмента для обнаружения уязвимостей был выбран сканер безопасности RedCheck, а также плагин для GLPI – FusionInventory, как инструмент для инвентаризации конечных устройств (АРМы, сетевое оборудование и т.д.).

Основная гипотеза заключается в формировании CPE-имен, либо их получения через API NVD на базе информации, полученной при помощи инструментов сбора данных, с целью их дальнейшего применения в контексте вышеописанной идеи.

Для сравнения FusionInventory был запущен в штатном режиме, RedCheck в режиме «Инвентаризация» с указанием всех дополнительных опций. Не смотря на разные цели инструментов, каждый из них выдал схожий набор данных, из которых наиболее важными для формирования CPE-имени являются имя и версия программного обеспечения. Таким образом, для формирования и поддержания в актуальном состоянии CPE-имен внутри системы управления активами возможно использовать лишь данные, полученные в результате инвентаризации.

Так как для получения CPE-имени требуется название программного обеспечения, в качестве примера рассмотрим пакет apache2 на Linux хосте. В данном случае поиск по ключевым словам не дает результата, так как CPE-имя, описывающее его (cpe:2.3:a:apache:http_server:2.0.9:*:*:*:*:*:*:*), в своем описании содержит «Apache HTTP Server». По итогу невозможен ни поиск по ключевым словам ни прямой поиск по CPE-именам, так как «apcache» в рамках CPE-имени является описанием вендора.

Потенциально существует возможность получения валидных CPE-имен посредством обращения к API NVD, при условии предварительной обработки имен программного обеспечения, полученных в результате сбора данных, например, посредством парсинга с помощью регулярных выражений. Однако, из-за высокой степени уникальности получаемых имён программного обеспечения, данная задача является крайне трудоемкой. Сложность этой задачи затрудняет второй вариант получения CPE-имен из собранных данных, а именно их самостоятельную генерацию.

Вышеописанные проблемы определяют необходимость создания национальной системы именования программного обеспечения (далее НСИПО). Стоит отметить, что среди исследователей уже предпринимались попытки разработки универсальных систем для управления уязвимостями [8], что в очередной раз подтверждает актуальность рассматриваемой проблемы. Так, в соответствии с мето-

дическим документом «Регламент включения информации об уязвимостях программного обеспечения и программно-аппаратных средств в банк данных угроз безопасности информации ФСТЭК России» при регистрации нового идентификатора BDU предполагается заполнение полей «Наименование уязвимого программного обеспечения» и «Версия уязвимого программного обеспечения». НСИПО предполагается использование дополнительного уникального идентификатора, который мог бы объединить два вышеописанных поля. Таким образом, БДУ ФСТЭК сможет хранить не только универсальный идентификатор уязвимого программного обеспечения, но и полные данные, по которым этот идентификатор можно получить, что позволяет решить проблему с получением CPE-имени, рассматриваемую ранее.

Предполагается, что сбор данных о программном обеспечении и обогащение системы управления активами выполняет агент на конечном устройстве, в дальнейшем полученные данные применяются в приложении сканере. Общий алгоритм работы системы приведен на рис. 1. Получение данных из НСИПО предполагается посредством API. Для поддержания сведений, хранящихся в системе управления активами в актуальном состоянии, задача данного агента должна выполняться периодически. Важно отметить, что в данном случае процесс пополнения системы управления активами данными является ответственностью специалистов по защите информации на местах, так как структура хранения данных может изменяться в зависимости от применяемой системы управления активами. Аналогичная ситуация существует и вопросе взаимодействия приложения сканера с системой управления активами.

Далее информация, полученная из НСИПО, используется для процесса управления уязвимостями, а именно для проведения пассивного сканирования на наличие уязвимостей в инфраструктуре, посредством отдельного приложения, в качестве ближайшего аналога которого можно привести приложение ScanOVAL.

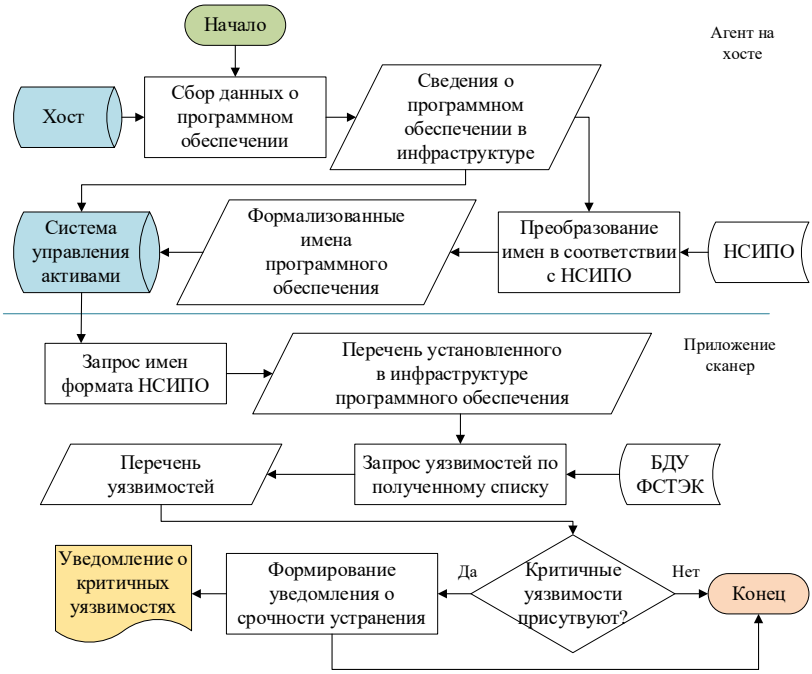


Рис. 1. Общий алгоритм работы НСИПО

Данное приложение так же может выполняться периодически, тем самым пополняя систему управления активами данными об уязвимостях, которые присутствуют в инфраструктуре. Так же используя метод автоматизации из более ранней работы [9], можно обогатить записи об уязвимостях для их лучшей интеграции с методическим документом ФСТЭК «Методика оценки уровня критичности уязвимостей программных, программно-аппаратных средств». Исходя из логики работы приложения можно выделить ряд достоинств:

1) Соответствие показателям эффективности, введенных регулятором. Так, например, при периодическом опросе БДУ ФСТЭК – раз в час, в худшем случае срок получения сведений о критичных уязвимостях программного обеспечения имеющегося в целевой инфраструктуре будет составлять не более часа, что оставляет 23 часа на устранение критичных уязвимостей.

2) Наличие подобного приложения не требует траты средств на специализированные решения, такие как сканеры безопасности или решения класса Vulnerability Management, так как для функционирования требуется лишь система менеджмента активов, в качестве которой может выступать решение с открытым исходным кодом, например, ранее упомянутый GLPI или отечественный аналог. Фактически приложение открывает возможность построения процесса управления уязвимостями для организаций с низким уровнем зрелости ИБ или со значительно ограниченными ресурсами, например, муниципальным учреждениям.

3) Введения универсального идентификатора программного обеспечения может значительно упросить процесс машинного обмена информацией об уязвимостях, в качестве примера такого обмена можно привести процесс уведомления НКЦКИ о произошедшем инциденте.

4) Потенциально возможно частичное решение проблемы с программным обеспечением собственной разработки [10], так как система управления активами может хранить сведения о зависимостях, применяемых в программном обеспечении собственной разработки, а в следствии и данные об уязвимостях в этих зависимостях.

Заключение

Проблема однозначной идентификации программного обеспечения является значимой и актуальной, так как ее решение может значительно повлиять на существующие сценарии реализации процесса управления уязвимостями. Существующие решения, в частности CPE, как наиболее успешный вариант, хоть и несли в себе идею их масштабного применения с течением времени пришли к тому, что они либо не применимы, либо могут лишь частично покрыть необходимость специалистов. Местами некорректная эксплуатация CPE привела к значительному усложнению ее практического применения. Создание национальной системы именования программного обеспечения позволит создать единый информационный базис для разработчиков программного обеспечения (системного и прикладного) и средств защиты информации, интеграторов и лицензиатов в области защиты информации, а также организаций, которые осуществляют их эксплуатацию. А интеграция национальной системы именования программного обеспечения в БДУ

ФСТЭК приведет к практической унификации процесса управления уязвимостями и возможности машинного взаимодействия всех заинтересованных сторон, что существенно повысит защищенность информационных систем. Работы в этом направлении соответствуют концепции импортозамещения и могут привести к тому, что БДУ ФСТЭК станет одним из мировых лидеров по поставке высококачественных данных об уязвимостях информационных систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Методический документ ФСТЭК от 17 мая 2023 г. «Руководство по организации процесса управления уязвимостями в органе (организации)» [Электронный ресурс]. – URL: https://fstec.ru/files/1096/---17--2023-/2011/---17--2023-.pdf (дата обращения: 10.05.2024).
2. Computer Security Incident Handling Guide [Электронный ресурс]. – URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf (дата обращения: 10.05.2024).
3. *Селигеев С.В., Жуков В.Г.* О проблеме верификации уязвимостей в публичных базах знаний // Решетневские чтения: Материалы XXVII Международной научно-практической конференции, посвященной памяти генерального конструктора ракетно-космических систем академика М.Ф. Решетнева, Красноярск, 08–10 ноября 2023 года. – Красноярск: Федеральное государственное бюджетное образовательное учреждение высшего образования "Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева", 2023. – С. 399-400. – EDN BYUGPC.
4. Official Common Platform Enumeration (CPE) Dictionary [Электронный ресурс]. – URL: https://nvd.nist.gov/products/cpe (дата обращения: 10.05.2024).
5. Security Content Automation Protocol [Электронный ресурс]. – URL: https://csrc.nist.gov/projects/security-content-automation-protocol (дата обращения: 10.05.2024).
6. *Дорофеев А В., Марков А.С.* Применение отечественных технологий для мониторинга информационной безопасности в условиях импортозамещения // Защита информации. Инсайд. – 2023. – № 3 (111). – С. 20-26. – EDN FDPTDW.
7. *Кальченко Д.М., Заливин А.Н., Федоров А.В.* Анализ программных и программно-аппаратных средств для защиты информации в информационных системах органов государственной власти // Научный результат. Информационные технологии. – 2023. – Т. 8, № 4. – С. 3-11. – DOI: 10.18413/2518-1092-2023-8-4-0-1. – EDN DULATC.

8. *Горкавенко В.С., Ажмухамедов И.М.* Разработка программного обеспечения для централизованного устранения уязвимостей хостов под управлением Unix-подобных операционных систем // Прикаспийский журнал: управление и высокие технологии. – 2019. – № 2(46). – С. 135-143. – EDN GJUBGW.
9. *Жуков В.Г., Селигеев С.В.* Автоматизация оценки уязвимостей программных, программно-аппаратных средств в целевой информационной системе // Прикаспийский журнал: управление и высокие технологии. – 2023. – № 4 (64). – С. 16-25. – DOI 10.54398/20741707_2023_4_16. – EDN SRMLBD.
10. *Селигеев С.В., Жуков В.Г.* О проблеме управления уязвимостями в программном обеспечении собственной разработки // Решетневские чтения: Материалы XXVII Международной научно-практической конференции, посвященной памяти генерального конструктора ракетно-космических систем академика М. Ф. Решетнева, Красноярск, 08–10 ноября 2023 года. – Красноярск: Федеральное государственное бюджетное образовательное учреждение высшего образования "Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева", 2023. – С. 401-403. – EDN SGTSJB.

УДК 004.8+004.056

Г.Н. Шурховецкий, М.С. Жукова, Л.В. Аршинский

Россия, г. Иркутск, Иркутский государственный университет путей сообщения

ЗАДАЧИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В СИСТЕМАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА: УГРОЗЫ И РЕШЕНИЯ

В исследовании изучаются вопросы информационной безопасности интеллектуальных информационных систем, которые используют модель предметной области, такую как база знаний для экспертных систем и архитектурный файл для искусственных нейронных сетей. Рассматривается защита команд, данных и коммуникаций. Вывод заключается в том, что в первую очередь необходимо защищать данные в форме модели предметной области, так как она является «интеллектуальным ядром» системы. Защита исполняемых файлов также важна, хотя и вторична. Защита коммуникации может осуществляться традиционными методами. Для защиты модели предлагается использовать метод фрагментации с последующим размещением фрагментов в разных, в том числе географически удалённых хранилищах.

Ключевые слова: *искусственный интеллект; информационная безопасность; экспертные системы; искусственные нейронные сети; метод рассечения-разнесения.*

The study examines the issues of information security of intelligent information systems that use a domain model, such as a knowledge base for expert systems and an architectural file for artificial neural networks. The protection of commands, data and communications is considered. The conclusion is that, first of all, it is necessary to protect data in the form of a domain model, since it is the "intellectual core" of the system. Protecting executable files is also important, although secondary. Communication protection can be carried out using traditional methods. To protect the model, it is proposed to use the fragmentation method with subsequent placement of fragments in different, including geographically remote repositories.

Keywords: *artificial intelligence; information security; expert systems; artificial neural networks; dissection-separation method.*

Введение

Методы и технологии искусственного интеллекта (ИИ) – актуальный тренд современного технологического развития, хотя уже сейчас ожидается очередная «инвестиционная зима» в сфере ИИ, тем не менее ИИ будет, как и технологии Bitcoin и стартапов занимать свою специализированную нишу, где они будут выполнять высокоэффективно свои задачи. Данные технологии появились в середине 40-х гг XX века именно как «умные алгоритмы» для решения в первую очередь военных задач по наведению на цель, анализ ситуации и т.д., они продолжают сохранять к себе интерес вплоть до сегодняшнего дня. Несмотря на то что повсеместное использование ИИ не требуется, существует ряд важных направлений, где их использование оправдано. Среди востребованных направлений – вопросно-ответные системы (в первую очередь чат-боты (например, для пожилых людей), особенно – основанные на больших лингвистических моделях), автоматизированные системы управления на предприятиях (АСУ и АСУТП), робототехника, медицина, сельское хозяйство, транспорт, государственное и муниципальное управление, искусство, военное дело и многое другое [1–4].

Пройдя через несколько этапов «инвестиционных зим» [5], технологии ИИ созрели до степени признания на законодательном уровне [6, 7].

В [5] ИИ определён как «комплекс технологических решений, позволяющий имитировать когнитивные функции человека (включая самообучение и поиск решений без заранее заданного алгоритма) и получать при выполнении конкретных задач результаты, сопоставимые, как минимум, с результатами интеллектуальной деятельности человека. Но здесь важно подчеркнуть, что речь идёт об имитации и это возможно только при наличии цифры. Например, Шебаршин Л.В. в книге «Из жизни начальника разведки» пишет следующее: «Роль ошибки, просчёта, легкомыслия и просто глупости никогда не учитывается в анализе политических ситуаций. В материалах расследований, отчётах, публицистических статьях, научных трудах… логика и разум вносятся туда, где господствовали неразбериха и некомпетентность, отметается элемент случайного, все события нанизываются на железный стержень рациональной, злой или доброй,

воли. В жизни так не бывает» [8]. Т.е. неизвестно как поведёт себя ИИ, если он столкнётся с реальной жизнью, а не цифрой. Комплекс технологических решений включает в себя информационно-коммуникационную инфраструктуру (в том числе информационные системы, информационно-телекоммуникационные сети, иные технические средства обработки информации), программное обеспечение (в том числе, в котором используются методы машинного обучения), процессы и сервисы по обработке данных и поиску решений».

В литературе и просто в общественном мнении широко обсуждаются различные риски, которые влечёт ИИ. Это прежде всего связано с непониманием как устроен ИИ, по каким принципам он функционирует, чем руководствуются его создатели [7–11].

В стереотипах, внедрённых современной культурой, активно живут представления об угрозе «захвата» и «порабощения» человечества определённой сверхсистемой ИИ, которая преследует свои, не относящиеся к людям цели и интересы (это является важнейшей особенностью настоящего интеллекта, преследование своих целей).

Все перечисленные риски и угрозы так или иначе относятся к влиянию ИИ на людей, хотя некоторые из них носят скорее газетно-публицистический, чем реальный характер [11]. В то же время существует и обратная, причём гораздо более актуальная угроза, – влияние людей на ИИ, – злонамеренные воздействия на системы ИИ для достижения своих, вполне человеческих целей. То есть угроза не людям со стороны ИИ, а угроза людям со стороны других людей через воздействие на системы ИИ. Здесь ожидается переход деятельности криминального хакинга и противодействия со стороны ИБ на скоростях суперкомпьютеров, а затем и квантовых вычислений, где ИИ будет играть ключевую роль [9].

Классы систем ИИ и их уязвимости

Системы искусственного интеллекта (СИИ) включают в себя различные программные продукты, такие как искусственные нейронные сети (ИНС), системы, основанные на знаниях (СОЗ), интеллектуальные информационно-поисковые системы, многоагентные системы и другие [12, 13]. Потенциальные уязвимости технологий ИИ связаны с информационной безопасностью, которая обеспечива-

ет защиту информационных ресурсов от внутренних и внешних угроз [14]. Предельно упрощая, можно ИБ свести к защищённости программной реализации СИИ.

Всякая (компьютерная) программа состоит из команд и данных. Команды – активная, данные – пассивная составляющая программного обеспечения (ПО). Также на сегодняшний день взаимодействие программ или их компонентов в глобальной сети или сети предприятия осуществляется посредством коммуникаций. Проблема делится на три составляющие: защита команд, данных и коммуникаций (рис. 1).

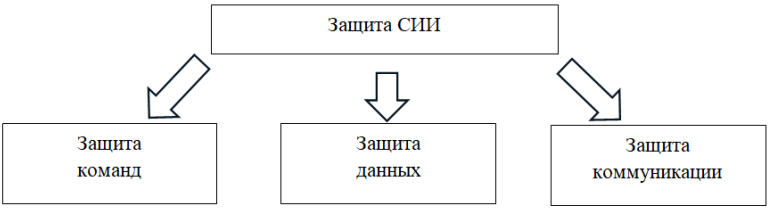


Рис. 1. Защита СИИ

Защита команд. Вторжение в систему команд – сложная проблема в области компьютерной безопасности. Такое вмешательство позволяет контролировать и изменять поведение системы, но требует глубокого понимания ПО, анализа кода и уверенности в отсутствии обновлений.

Теоретически можно заменить секцию импорта исполняемого файла, но нужно убедиться, что программа продолжит работать без доступа к ресурсам. Пример деструктивного воздействия – компьютерные вирусы, которые остаются незамеченными как можно дольше, не нарушая основную функциональность программы.

Сложные атаки затрудняются сжатием и шифрованием команд, но такое вмешательство быстро обнаруживается. Изменение системы команд увеличивает эффективность атаки, но в долгосрочной перспективе этот подход не оптимален.

Защита данных. Данные являются неотъемлемой частью программного обеспечения. Интерес представляют атаки, когда данные хранятся отдельно от исполняемого файла, например, в базах данных и

внешних файлах. В СИИ применяется принцип внешнего хранения данных, где вмешательство в базу знаний (БЗ) или коэффициенты искусственных нейронных сетей (ИНС) искажает модель предметной области (ПрО) и может привести к неправильным решениям, классификации и ответам на вопросы. Возможные последствия включают ошибки в критически важных технологиях и работе экспертных систем (ЭС).

Общая структура СИИ, основанных на моделях ПрО, показана на рис. 2.

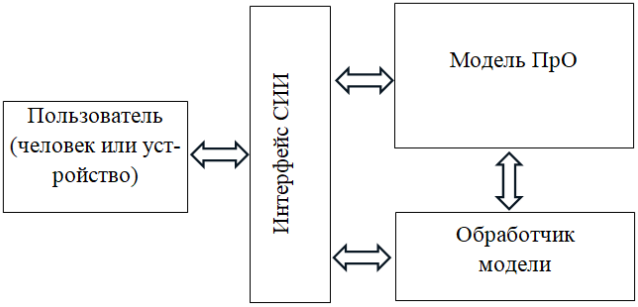


Рис. 2. Общая структура СИИ

Обработчик модели – универсальный модуль, работа которого не зависит от модели, но на неё влияет. В ЭС это решатель, в ИНС – обработчик связей. Таким образом, вмешательство в базу знаний ЭС или коэффициенты ИНС означает вмешательство в модель предметной области и её искажение. Если провести аналогию с человеческим интеллектом, искажение модели будет равносильно разрушению или нарушению представлений человека о мире и его когнитивных способностей. В СИИ это может привести к неправильному решению задач, например, неверным решениям (в АСУТП), неправильной классификации или идентификации образов (распознавание образов, ИНС), неправильным ответам на вопросы (ЭС). Легко представить возможные последствия, если речь идёт о вмешательстве в АСУТП для критически важной технологии или в работу ЭС, когда её рекомендации становятся основой для принятия важных решений.

Преднамеренное искажение модели для обмана системы – интересная тема. В ИНС нужно учитывать взаимосвязи и веса входных нейронов, а также понимать, как сеть обрабатывает информацию.

Злоумышленник может повредить файл, создать свою сеть и заменить исходный файл. В системах на основе знаний можно изменять фрагменты БЗ, что незаметно для пользователя. Атака отравлением данных (Data Poisoning) снижает точность обучения [15–17].

Стоит отметить, что вмешательство в данные интеллектуальных систем имеет интересную особенность: согласно источнику [18] и ряду других источников, знания (модель ПрО), хранящиеся в интеллектуальной системе, включают в себя характеристики как данных, так и команд. Этот феномен называется активностью знаний. Если знания размещены во внешнем файле, то воздействие на этот файл похоже на воздействие на систему команд, что предоставляет злоумышленнику дополнительные возможности, недоступные при работе с обычным ПО.

Защита коммуникации. Защита коммуникации – ключевой аспект безопасности интеллектуальных информационных систем. Современные подходы включают сервис-ориентированные архитектуры, распределённое хранение данных и облачные вычисления. Обмен данными между удалёнными компьютерами является неотъемлемой частью функционирования ИС, включая СИИ. Также системы ИИ сами могут быть частью системы хранения данных [19]. Основные цели атаки на коммуникацию: нарушение целостности и доступности информации, а также конфиденциальности [20]. Достижение этих целей возможно разными способами, такими как внедрение нерегламентированных возможностей, перехват трафика и изменение параметров системы [20, 21]. Проблема заключается в «расширенной постоянной угрозе», контроле функционирования СИИ и угрозах, связанных с развитием сетевых технологий.

Вообще, развитие и усложнение сетевых технологий могут порождать и порождают новые угрозы, которые не всегда можно принять во внимание (угрозы нулевого дня).

Меры противодействия

Злоумышленники могут получить доступ к данным через вредоносные файлы или компьютерные сети. Не все данные могут быть защищены постоянно, особенно если они размещены в интернете. В этом тексте обсуждаются данные, используемые в СИИ, и рассматриваются две категории нарушителей: внешние и внутренние.

Внешний нарушитель. Шифрование – основной метод защиты от внешних угроз. Атакующий не сможет целенаправленно изменить модель ПрО, если не поймёт её семантику. Угрозу можно предотвратить с помощью резервного копирования модели. Восстановление системы после сбоя происходит быстро, поэтому долгосрочное вмешательство маловероятно. Тактика разрушения оправдана только при быстрых и фатальных последствиях атаки, например, разрушении АСУТП для критически важных производств [20]. Если СИИ не подключена к глобальной сети, получить постоянный доступ извне сложно, но при взаимодействии с глобальными сетями возможны атаки [22]. Для противодействия рекомендуется использовать защищённые интернет-протоколы, технологии VPN и другие инструменты. Также необходимо применять традиционные средства защиты внутренней сети предприятия.

Интересным приёмом защиты внешних данных может быть разделение критической информации и разнесение её по разным местам хранения (хранилищам) [19] (рис. 3).

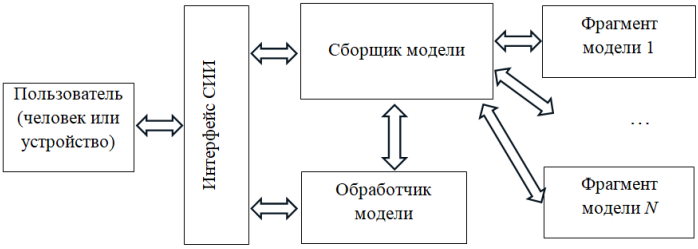


Рис. 3. СИИ с распределённым, в т.ч. географически распределённым хранением модели ПрО

Распределённое и децентрализованное хранение данных затрудняет поиск для злоумышленников, особенно если данные зашифрованы. Более высокий уровень защиты достигается с использованием побитового рассечения в алгоритме, представленном в работе [23], что делает невозможным извлечение информации за разумное время без криптографии. Хотя криптография важна, она становится менее критичной и дополнительно усиливает защиту. Расшифровка данных, собранных из всех хранилищ (атака «злоумышленник обладает всеми потоками»), не даст результата. Только внутренний на-

рушитель (ограниченный круг лиц) или внедренное программное обеспечение, получившее доступ к описанию процесса сборки при неправильном хранении, смогут восстановить исходный файл.

Внутренний нарушитель. Первостепенная мера для защиты от внутренних угроз также традиционна – это разделение доступа. Для критически важных ИС и СИИ требуется разграничение не только логического, но и физического доступа. Ограничение круга лиц, имеющих доступ к работе с СИИ и особенно к изменению модели ПрО, должно быть неотъемлемой частью безопасности. В этом нет ничего специфичного для ИС и СИИ. При необходимости можно использовать полный спектр мер: контроль доступа лиц, контроль доступа к оборудованию, видеонаблюдение, защита серверных устройств, контроль доступа к вычислительной технике и другие методы.

Заключение

В результате проведённого анализа можно сделать предварительные выводы:

I. В первую очередь необходимо защищать модель ПрО – «интеллектуальное ядро» системы, то есть базу знаний для экспертных систем и архитектурный файл для искусственных нейронных сетей. Обучающие выборки требуются для ИНС и систем распознавания образов.
II. Интересный метод защиты «интеллектуального ядра» – его распределённое хранение на разных серверах, в том числе географически удалённых, местоположение которых неизвестно нарушителю. Фрагменты следует шифровать или использовать другие методы защиты для обеспечения сопоставимой стойкости.
III. Эффективный метод распределённого хранения – побитовое рассечение-разнесение. Даже без шифрования доступ ко всем фрагментам ядра практически не оставляет злоумышленнику шансов получить оригинальную информацию за приемлемое время, даже если фрагменты передаются по открытым каналам. Шифрование ещё больше ухудшает его положение.
IV. Защита исполняемых файлов также важна, хотя и вторична.
V. Защита коммуникации может осуществляться традиционными средствами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Лапушкин А.* Сферы применения систем искусственного интеллекта. – URL: https://maff.io/media/sfery-primeneniya-sistem-iskusstvennogo-intellekta/ (дата обращения: 21.05.2024).
2. Сферы применения искусственного интеллекта: от медицины до сельского хозяйства. – URL: https://gb.ru/blog/sfery-primeneniya-iskusstvennogo-intellekta/ (дата обращения: 21.05.2024).
3. На что способен искусственный интеллект сегодня и каков его потенциал. – URL: https://trends.rbc.ru/trends/industry/cmrm/619766d59a79471862e77e8a (дата обращения: 21.05.2024).
4. 5 применений ИИ, в которых он конкурирует с человеком. – URL: https://habr.com/ru/companies/toshibarus/articles/580930/ (дата обращения: 21.05.2024).
5. Путь искусственного интеллекта от фантастической идеи к научной отрасли. – URL: https://habr.com/ru/companies/cloud4y/articles/469447/ (дата обращения: 21.05.2024).
6. Федеральный закон от 24.04.2020 N 123-ФЗ. «О проведении эксперимента по установлению специального регулирования в целях создания необходимых условий для разработки и внедрения технологий искусственного интеллекта в субъекте Российской Федерации - городе федерального значения Москве и внесении изменений в статьи 6 и 10 Федерального закона «О персональных данных». – URL: http://publication.pravo.gov.ru/Document/View/ 0001202004240030 (дата обращения: 21.05.2024).
7. *Ли Яо.* Нормативно-правовое регулирование генеративного искусственного интеллекта в Великобритании, США, Европейском союзе и Китае // Право. Журнал Высшей школы экономики, 2023. – Т. 16, № 3. – С. 245-267.
8. *Шебаршин Л.В.* Из жизни начальника разведки. – М.: Терра – Книжный клуб, 1997. – 192 с.
9. *Шнайер Б.* Взломать всё. Как сильные мира сего используют уязвимости систем в своих интересах. – М.: Изд-во Альпина Паблишер, 2023. – 376 с.
10. *Морхат П.М.* Риски и угрозы, связанные с применением искусственного интеллекта // Аграрное и земельное право. – 2017. – № 12 (156). – С. 60-65.
11. *Артамонов В.А., Артамонова Е.В.* Проблемы искусственного интеллекта: мифы и реальность. – URL: https://cyberleninka.ru/article/n/problemy-iskusstvennogo-intellekta-mify-i-realnost/viewer (дата обращения: 21.03.2024).
12. *Рассел С., Норвиг П.* Искусственный интеллект: современный подход. – 2-е изд.: пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1408 с.
13. *Аршинский Л.В., Жукова М.С.* Интеллектуальные информационные системы и технологии: учебное пособие. – Иркутск: ИрГУПС, 2023. – 128 с.

14. *Вострецова Е.В.* Основы информационной безопасности: учебное пособие для студентов вузов. – Екатеринбург: Изд-во Урал. Ун-та, 2019. – 204 с.
15. *Намиот Д.Е.* Введение в атаки отравлением на модели машинного обучения // International Journal of Open Information Technologies. – 2023. – Т. 11, № 3. – С. 58-66.
16. *Bagdasaryan E., Shmatikov V.* Blind backdoors in deep learning models // 30th USENIX Security Symposium. – 2021. – P. 1505-1521.
17. *Апресов С.* Что такое «отравление данных». Методы защиты от атак data poisoning. – URL: https://digitalocean.ru/n/podryvnaya-deyatelnost/ (дата обращения: 28.03.2024).
18. *Гаскаров Д.В.* Интеллектуальные информационные системы: учеб. для вузов. – М.: Высш. шк., 2003. – 431 с.
19. *Жилин В.В., Сафьян О.А.* Искусственный интеллект в системах хранения данных // Вестник Донского государственного технического университета. – 2020. – Т. 20, № 2. – С. 196-200.
20. *Бабенко Г.В.* Анализ современных угроз безопасности информации, возникающих при сетевом взаимодействии // Вестник АГТУ. Сер.: Управление, вычислительная техника и информатика, 2010. – № 2. – URL: https://cyberleninka.ru/article/n/analiz-sovremennyh-ugroz-bezopasnosti-informatsii-voznikayuschih-pri-setevom-vzaimodeystvii/viewer (дата обращения: 28.03.2024).
21. *Вартанян А.А.* Угрозы и атаки сетевой безопасности. – URL: https://cyberleninka.ru/article/n/ugrozy-i-ataki-setevoy-bezopasnosti/viewer (дата обращения: 28.03.2024).
22. *Ромашкина Н.П., Махукова А.В.* Компьютерная вредоносная атака на ядерную программу Ирана // Информационные войны. – 2013. – № 4. – С. 40-50.
23. *Аршинский Л.В., Шурховецкий Г.Н.* Особенности применения метода рассечения-разнесения для безопасного хранения данных во внешних хранилищах // Информационные технологии. – 2021. – № 5. – Т. 27. – С. 259-266. – DOI: 10.17587/it.27.259-266.

УДК 004.056

Ю.К. Язов, А.П. Панфилов, М.А. Тарелкин

Россия, г. Воронеж, Государственный научно-исследовательский испытательный институт ФСТЭК России

СОСТАВНЫЕ СЕТИ ПЕТРИ-МАРКОВА И ИХ ПРИМЕНЕНИЕ ДЛЯ МОДЕЛИРОВАНИЯ УГРОЗ БЕЗОПАСНОСТИ ИНФОРМАЦИИ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Цель данной статьи состоит в раскрытии возможностей нового аппарата составных сетей Петри-Маркова для моделирования процессов реализации угроз безопасности информации в информационных системах. Отмечаются особенностей данного аппарата и его отличия от традиционного аппарата сетей Петри-Маркова. Указываются новые возможности для моделирования, предоставляемые аппаратом составных сетей. Приводятся основные правила построения составных сетей Петри-Маркова и раскрываются особенности расчета вероятностно-временных характеристик срабатывания сетей.

Ключевые слова: *составная сеть; парциальный процесс; логический переход; состояние; логическое условие; динамика реализации; вероятность; математическое ожидание.*

The goal of article consists in disclosure the capabilities of the new apparatus of composite Petri-Markov networks to model the processes of implementing information security threats in information systems. The features of this apparatus and its differences from the traditional apparatus of Petri-Markov networks are noted. New modeling capabilities provided by the apparatus of composite network are indicated. Basic rules to construct composite Petri-Markov networks are given. Calculation features of probabilistic-time characteristics of networks actuation are disclosed.

Keywords: *composite network; partial process; logical transition; state; logical condition; implementation dynamics; probability; mathematical expectation.*

В соответствии с действующими нормативными правовыми документами по технической защите информации в информационных системах (ИС) проводится анализ угроз безопасности информации. Сегодня такой анализ направлен на оценку рисков реализации

угроз (возможностей реализации и ожидаемого ущерба) и проводится экспертным путем, но основной тенденцией его совершенствования является переход от качественных к количественным методам с разработкой соответствующих моделей для оценки возможности реализации угроз. Сегодня для этого разработано много как аналитических, так и имитационных моделей. Однако в связи со значительной трудоемкостью разработки и узким назначением имитационных моделей основное внимание уделяется аналитическим моделям. Вместе с тем, аналитические модели направлены, в основном, на оценку возможности реализации угроз. Во многих из них, основанных, например, на теории графов, формальных моделях безопасности, исчислении высказываний и теории предикатов, сетях Петри и др., не учитывается фактор времени. Вместе с тем при решении многих задач такое игнорирование невозможно, так как приводит к некорректным выводам. Особенно это важно, когда оценивается возможность опережения и своевременного блокирования мерами защиты процесса реализации угроз безопасности информации в ИС. Однако случайный характер процесса реализации угроз и сложность определения вероятностно-временных характеристик, определяющих динамику реализации угроз, обусловливает необходимость разработки для этого соответствующих аналитических моделей процессов реализации угроз. Первоначально для этих целей использовался аппарат марковских, затем полумарковских процессов с непрерывным временем и дискретными состояниями. Однако применение аппарата марковских процессов для реализации угроз сдерживается тем, что невозможно:

- получать в общем случае аналитические решения системы дифференциальных уравнений, описывающих моделируемый процесс, когда количество его состояний более 4. Приходится применять численные методы решения;
- моделировать параллельные взаимосвязанные процессы;
- вводить в систему дифференциальных уравнений логические условия, которые часто имеют место при реализации угроз безопасности информации.

Для обеспечения моделирования параллельных взаимосвязанных процессов в 1954-1955 гг. независимо П. Леви, В. Смитом и Л. Такачем был предложен аппарат полумарковских процессов (ПМП). Однако он тоже имеет определенные ограничения, такие как:

- невозможность учета логических условий;
- сложность моделирования циклических операций с задаваемыми условиями начала и завершения их выполнения;
- наличие только одного начального состояния процесса, в то время как могут иметь место несколько состояний, из которых может начинаться процесс с определенными, как правило, случайными задержками во времени и др.

Вместе с тем, благодаря появлению аппарата ПМП, стал возможным переход к сетям Петри-Маркова (СПМ), которые впервые были предложены в [1] для решения задач математического моделирования отказов/восстановлений технических систем в рамках теории надежности (далее – традиционные СПМ). Суть предложения заключалась в симбиозе аппарата сетей Петри и аппарата ПМП для описания вероятностно-временных характеристик моделируемых процессов[1].

Однако аппарат ПМП не позволял моделировать процессы с логическими условиями выполнения. В связи с этим в [2] был предложен подход к аналитическому моделированию параллельных асинхронных процессов с логическими условиями выполнения, основанный на составных сетях Петри-Маркова (ССПМ). Под ССПМ понимается взаимосвязанная совокупность сетевых фрагментов, каждый из которых представляет собой марковский или полумарковский процесс, объединяемых («сшиваемых») так называемыми логическими переходами, срабатывающими по определенным логическим правилам.

Так же, как сети Петри и традиционные СПМ, ССПМ представляются в виде ориентированного графа, где каждая позиция обозначена кружком с номером, переход – жирной чертой (для логического перехода) или нежирной чертой (для простого перехода) с номером и буквой z, а направления перемещения процесса по графу

[1] Следует подчеркнуть отличие аппарата СПМ от аппарата сетей Петри, которые изобретены в августе 1939 года Карлом Адамом Петри для моделирования химических процессов, а впервые применительно к информационным потокам были описаны в его диссертации в 1962 г. Этот аппарат позволяет в аналитическом виде получать соотношения для расчета вероятностно-временных характеристик моделируемых случайных процессов, а аппарат сетей Петри ориентирован на имитационное моделирование таких процессов.

показываются стрелками. В начальную позицию устанавливается маркер, которым отмечается перемещение моделируемого процесса из состояния в состояние. При этом перемещение процесса из позиции в переход осуществляется за конечное случайное время, а из перехода в позицию – мгновенно с единичной вероятностью. Пример графа ССПМ приведен на рис. 1.

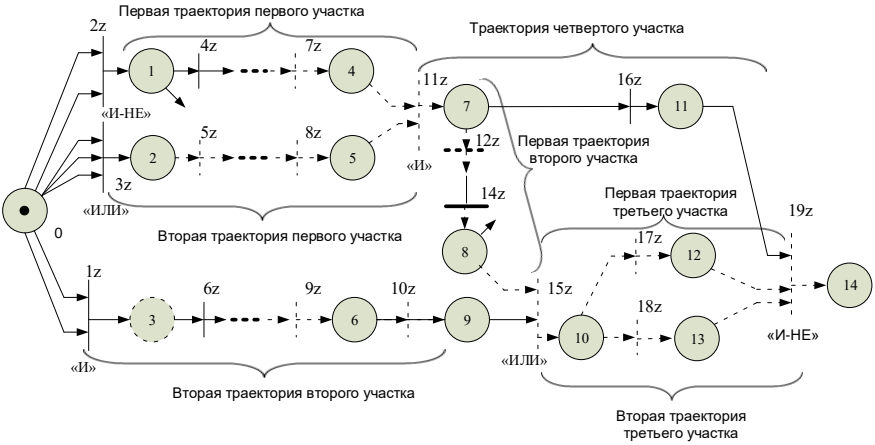


Рис. 1. Пример графа составной сети Петри-Маркова

ССПМ существенно отличаются от традиционных СПМ:

- традиционные СПМ являются сетями, в которых переходы обязательно срабатывают, если для каждой входящей в переход дуги имеется не менее одного маркера, однако в ССПМ для перехода может быть определена иная логика, с которой должны выполняться заданные для логического перехода логические условия (например, условие, когда переход срабатывает, если по одной входящей дуге парциальный процесс подошел к переходу, а по другой входящей дуге парциальный процесс запоздал);
- каждая ССПМ состоит из участков, соединяемых между собой через логические переходы, при этом на каждом участке моделируемый процесс может быть как марковским, так и полумарковским, чего нет в традиционных СПМ.

Порядок построения ССПМ состоит в следующем:

1. Определяются все состояния, в которых может находиться моделируемый процесс, и логические переходы, в которых чаще всего используются логические условия, примеры которых приведены в табл. 1.

2. Определяются простые переходы для каждого участка ССПМ: между начальным состоянием и первым логическим переходом, между логическими переходами и, наконец, между последним логическим переходом и конечным состоянием. Все состояния и переходы нумеруются.

Таблица 1

Примеры логических переходов и соотношения для расчета математических ожиданий времен их срабатывания

Логическое условие	Соотношение для расчета математического ожидания времени срабатывания перехода
$1 \wedge 2$ (условие «И»)	$\overline{\tau_{"И"}} = \left(\overline{\tau_1}^2 + \overline{\tau_1} \cdot \overline{\tau_2} + \overline{\tau_2}^2\right) \Big/ \left(\overline{\tau_1} + \overline{\tau_2}\right)$, где $\overline{\tau_1}$ и $\overline{\tau_2}$, математические ожидания времени поступления на логический переход парциальных процессов по двум дугам
$1 \vee 2$ (условие «ИЛИ»)	$\overline{\tau_{"ИЛИ"}} = \left(\overline{\tau_1} \cdot \overline{\tau_2}\right) \Big/ \overline{\tau_1} + \overline{\tau_2}$
$(1 \wedge 2) \vee 3$ (условие «И-ИЛИ»)	$\overline{\tau_{"И-ИЛИ"}} = \overline{\tau_1} + \overline{\tau_2} - \frac{\overline{\tau_1} \cdot \overline{\tau_2}}{\overline{\tau_1} + \overline{\tau_2}} - \frac{\overline{\tau_1}^2 \cdot \overline{\tau_2}}{\left(\overline{\tau_1} + \overline{\tau_2}\right)^2} - \frac{\overline{\tau_3}^2 \cdot \overline{\tau_1}}{\left(\overline{\tau_1} + \overline{\tau_3}\right)^2} - \frac{\overline{\tau_3}^2 \cdot \overline{\tau_2}}{\left(\overline{\tau_2} + \overline{\tau_3}\right)^2} + \frac{}{\overline{\tau_1} \cdot}$
$1 – 2$ (условие «И-НЕ»)	$\overline{\tau_{"И-НЕ"}} = \overline{\tau_1} \cdot \left(1 + \frac{\overline{\tau_1}}{\overline{\tau_2}}\right)$
$1 \vee 2 – 3$ (условие «ИЛИ-НЕ»)	$\overline{\tau_{"ИЛИ-НЕ"}} = \overline{\tau_{"ИЛИ"}^{(1,2)}} \cdot \left(1 + \frac{\overline{\tau_{"ИЛИ"}^{(1,2)}}}{\overline{\tau_3}}\right)$, где $\overline{\tau_{"ИЛИ"}^{(1,2)}} = \overline{\tau_1} \cdot \overline{\tau_2} \Big/ \left(\overline{\tau_1} + \overline{\tau_2}\right)$

3. Для каждого логического перехода формируются входные и выходные дуги, соединяющие переход с инцидентными позициями, и строится граф ССПМ.

4. Осуществляется разметка ССПМ путем помещения маркера в начальную позицию (или начальные позиции, если их несколько).

5. Каждому перемещению из состояния в переход ставится в соответствие значение математического ожидания времени такого перемещения.

При построении графа, когда парциальный процесс является *марковским,* должны выполняться следующие правила:

- если две или большее количество дуг входят в переход, то этот переход является логическим, несколько дуг не могут входить в простой переход;
- из позиции не должно выходить более одной дуги;
- не допускаются ситуации, чтобы дуга выходила из позиции и входила в позицию, или выходила из перехода и входила в переход;
- количество дуг, выходящих из перехода, определяется составом и содержанием моделируемых действий. При количестве выходящих из перехода дуг две и более, по сути, имеет место разветвление процесса с его перемещением к разным позициям. При этом осуществляется «размножение» маркера, и по всем исходящим из перехода дугам маркеры (по каждому маркеру на дугу) перемещаются в инцидентные позиции мгновенно с единичной вероятностью.

При построении графа для ССПМ на основе *полумарковских* парциальных процессов считается, что если в ССПМ хотя бы один из парциальных процессов является полумарковским (а остальные марковские), то ССПМ в целом моделирует полумарковский процесс. Для такой ССПМ должны выполняться указанные ранее правила построения ССПМ для марковских процессов, а также следующие:

- если из позиции выходят две дуги, то должны быть указаны вероятности выбора каждой дуги, по которой будет далее развиваться процесс;
- если в позицию входят две и более дуг, то рассматриваются отдельно альтернативные варианты перемещения парциальных процессов по соответствующим траекториям с определенной вероятностью выбора каждой траектории.

При расчете вероятностно-временных характеристик срабатывания ССПМ (математического ожидания времени срабатывания и вероятности срабатывания за заданное время) должны выполняться следующие основные правила:

1. Если участок между двумя логическими переходами имеет одну траекторию и имеются выходящие из нее две или более траектории, которые затем сходятся в логическом переходе (рис. 2), то сначала рассчитываются времена по каждой из траекторий, а затем рассчитывается время срабатывания логического перехода с учетом времен, рассчитанных для траекторий №2 и №3.

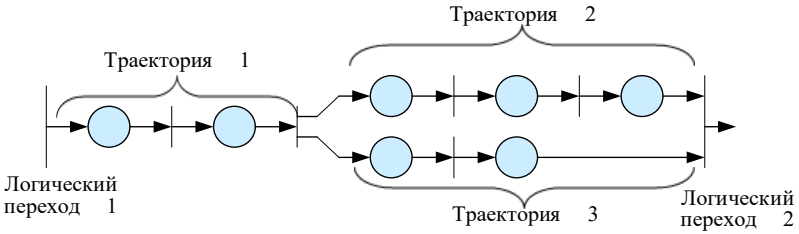


Рис. 2. Схема ССПМ с разветвляющейся траекторией

Например, если математические ожидания времен выполнения парциальных процессов по траекториям №1 – №3 равны соответственно $\overline{\tau_1}, \overline{\tau_2}$ и $\overline{\tau_3}$, а логический переход №2 срабатывает по логике «ИЛИ», то математическое ожидание времени $\overline{\tau}$ срабатывания логического перехода рассчитывается следующим образом:

$$\overline{\tau} = \overline{\tau_1} + \frac{\overline{\tau_2} \cdot \overline{\tau_3}}{\overline{\tau_2} + \overline{\tau_3}}. \qquad (1)$$

2. Если на двух участках траектории разветвляются, но первый участок замыкается на свой логический переход, а затем оба парциальных процесса приходят к одному и тому же логическому переходу (рис. 3), то расчет математического ожидания времени выполнения ССПМ (в данном случае времени срабатывания переход 7z) проводится следующим образом.

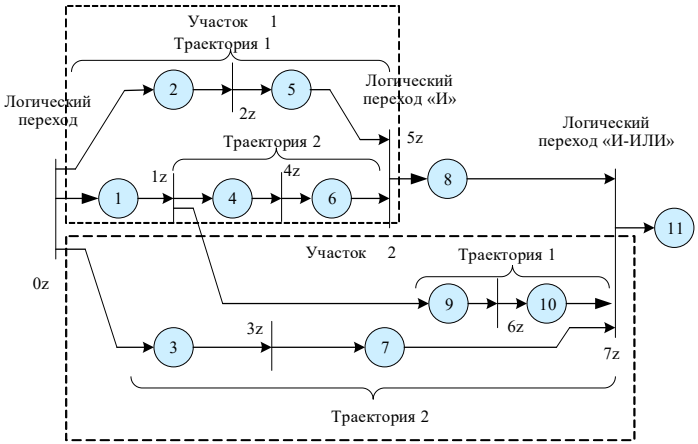


Рис. 3. Схема ССПМ в случае разветвления траекторий и замыкании их на разные логические переходы

Пусть математические ожидания времен выполнения парциальных процессов по траектории №1 и №2 первого участка[2] равны соответственно $\overline{\tau_1^{(1)}} = \overline{\tau_{22}} + \overline{\tau_{55}}$ и $\overline{\tau_2^{(1)}} = \overline{\tau_{11}} + \overline{\tau_{44}} + \overline{\tau_{65}}$, а по траекториям №1 и №2 второго участка – соответственно $\overline{\tau_1^{(2)}} = \overline{\tau_{11}} + \overline{\tau_{96}} + \overline{\tau_{10,7}}$ и $\overline{\tau_2^{(2)}} = \overline{\tau_{33}} + \overline{\tau_{77}}$, логический переход $5z$ имеет логику «И», а логический переход $7z$ – логику «$И_1$-$ИЛИ_2$», то есть при условии, что парциальный процесс подойдет к логическому переходу по дуге (8,7z) и хотя бы по одной из траекторий №1 и №2 второго участка.

Тогда математические ожидания времен срабатывания логических переходов $5z$ и $7z$ рассчитываются следующим образом:

[2] Здесь приняты обозначения $\overline{\tau_j^{(i)}}$ – математическое ожидание времени перемещения процесса по j-й траектории i-го участка; $\overline{\tau_{mn}}$ – времени перемещения процесса из m-ой позиции в n-й переход.

$$\overline{\tau_{5(z)}} = \frac{\overline{\tau_1^{(1)}}^2 + \overline{\tau_1^{(1)}} \cdot \overline{\tau_2^{(1)}} + \overline{\tau_2^{(1)}}^2}{\overline{\tau_1^{(1)}} + \overline{\tau_2^{(1)}}}; \tag{2}$$

$$\overline{\tau_{7(z)}} = \frac{\left(\overline{\tau_{5(z)}} + \overline{\tau_{8,7}}\right)^2 + \left(\overline{\tau_{5(z)}} + \overline{\tau_{8,7}}\right) \cdot \frac{\overline{\tau_1^{(2)}} \cdot \overline{\tau_2^{(2)}}}{\overline{\tau_1^{(2)}} + \overline{\tau_2^{(2)}}} + \left(\frac{\overline{\tau_1^{(2)}} \cdot \overline{\tau_2^{(2)}}}{\overline{\tau_1^{(2)}} + \overline{\tau_2^{(2)}}}\right)^2}{\overline{\tau_{7(z)}} + \overline{\tau_{8,7}} + \frac{\overline{\tau_1^{(2)}} \cdot \overline{\tau_2^{(2)}}}{\overline{\tau_1^{(2)}} + \overline{\tau_2^{(2)}}}}$$

Если моделируется процесс реализации угрозы в условиях применения мер (средств) защиты, то необходимо иметь в виду, что структура графа ССПМ даже при применении неэффективных мер (средств) защиты оказывается отличной от структуры графа ССПМ, соответствующего их отсутствию, так как затраты времени на выполнение действий по защите остаются.

Для расчета вероятностно-временных характеристик парциальных процессов при наличии средств защиты предлагается использовать аппарат вероятностного просеивания потоков [2, 3]. Например, когда в ИС используется мера аутентификации и установлена система обнаружения вторжений, математические ожидания времен подбора пароля и обхода системы обнаружения рассчитываются по формулам:

$$\overline{\tau_{passw}} = \overline{\tau_{symb}} \big/ p_{passw} \text{ и } \overline{\tau_u} = \overline{\tau_{analys}} \big/ \left(1 - p_{det}\right), \tag{3}$$

где $\overline{\tau_{symb}}$ и $\overline{\tau_{analys}}$ – математические ожидания времени набора символов пароля и времени анализа сигнатуры или трафика системой обнаружения;

p_{passw} и p_{det} – вероятности подбора пароля и обнаружения сетевой атаки.

Таким образом, аппарат ССПМ позволяет вместо качественного анализа перейти к количественным оценкам возможностей реализации угроз безопасности информации в ИС и, тем самым, существенно повысить достоверность результатов их анализа.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Игнатьев В.М., Ларкин Е.В*. Сети Петри-Маркова. – Тула: ТулГТУ, 1994. – 163 с.
2. *Язов Ю.К.* Основы методологии количественной оценки эффективности защиты информации в компьютерных системах. – Ростов-на-Дону: СКНЦ ВШ, 2006. – 274 с.
3. *Климов Г.П.* Стохастические системы обслуживания. – М.: Наука, 1966. – 244 с.

УДК 004.056

А.О. Яковлева, М.Н. Жукова

Россия, г. Красноярск, Сибирский государственный университет науки и технологий им. академика М.Ф. Решетнев

РАЗРАБОТКА ПРОГРАММНОГО РЕШЕНИЯ ДЛЯ ФОРМИРОВАНИЯ АДАПТИРОВАННОГО НАБОРА МЕР ПРИ ПРОВЕДЕНИИ ОЦЕНКИ СООТВЕТСТВИЯ ПО ГОСТ Р 57580.2 В ЗАВИСИМОСТИ ОТ ТРЕБОВАНИЙ ГОСТ Р 57580.1

В статье рассмотрены положения нормативно правовых актов Банка России устанавливающие требования для организаций кредитно-финансовой сферы. Основная задача заключается в формировании технического задания на создание программного решения, инструмента, позволяющего определить уровень защиты для контура безопасности и сформировать перечень мер, подлежащих оценке, для исполнения требований ГОСТ Р 57580.1 (и.2).

Ключевые слова: *контур безопасности; уровень защищенности; набор мер.*

The article considers the provisions of the regulatory legal acts of the Bank of Russia establishing requirements for organizations in the credit and financial sector. The main task is to form a technical specification for the creation of a software solution, a tool that allows you to determine the level of protection for the safety contur and form a list of measures to be evaluated to meet the requirements of of GOST R 57580.1 (and .2).

Keywords*: security contour; level of protection; set of measures.*

Введение

Вопрос результативности построения системы защиты с применением серии стандартов ГОСТ Р 57580 для кредитных организаций является сложным для изучения. В одной из последних статей [1], авторами был сформулирован вывод, что основная проблема создания и поддержания функционирования систем в соответствии со стандартами серии 57580 заключается в том, что процесс проведения оценки соответствия не сопровождается наличием дополнительных

методических рекомендаций и алгоритмов. Их наличие позволило бы кредитным организациям сформировать подход к выполнению требований, сделать его упорядоченным и понятным.

Но прежде чем выбрать меры защиты информации (ЗИ), реализовать их и провести оценку соответствия, следует решить следующие вопросы:

- какое количество контуров безопасности подлежит оценке;
- какие уровни ЗИ соответствуют выделенным контурам;
- какой набор мер подлежит оценке;
- как сформировать адаптированный набор мер.

Согласно требованиям, в финансовой организации формируются один или несколько контуров для которых может быть установлен разный уровень ЗИ.

Проблема проведения оценки соответствия по требованиям ГОСТ Р 57580.1 (и .2) заключается в том, чтобы верно определить количество контуров безопасности и требуемый уровень ЗИ для каждого контура, поскольку именно от этого зависит формирование базового и адаптированного набора оцениваемых мер. Схема проведения оценки представлена на рис. 1.

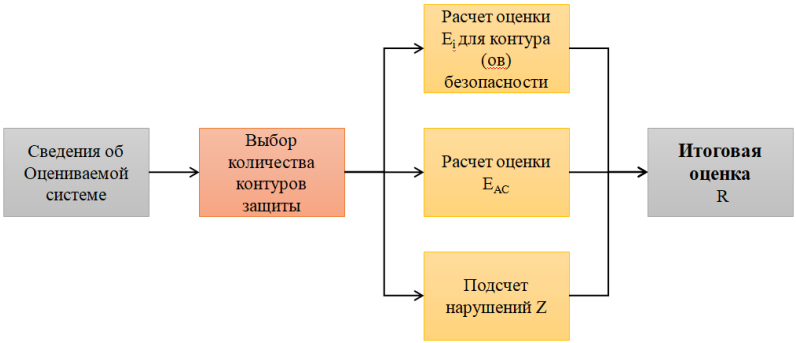


Рис. 1. Типовая схема проведения оценки

Таким образом, задача в целом сводится к разработке инструмента, позволяющего автоматизировано определить уровень ЗИ для контура безопасности и сформировать адаптированный перечень мер ЗИ, подлежащих оценке, что позволит снизить трудоемкость и сложность этого процесса.

Данный инструмент, планируемый в виде программного комплекса, должен формировать перечень мер ЗИ, подлежащих оценке для исполнения требований ГОСТ Р 57580.1 (и.2) в соответствии с положениями нормативно правовых актов (НПА) Банка России (БР) для организаций кредитно-финансовой сферы. Для разработки подобного решения в статье приводится, краткое содержание частей технического задания, описывающих основные проблемные моменты расчетов и оценки.

Формирование перечня выполняется поэтапно, по следующему алгоритму:

1. Аудит системы по требованиям НПА БР и серии ГОСТов:
   - определение количества контуров безопасности;
   - определение уровня ЗИ для каждого контура при оценке;
2. Формирование перечня мер для оценки:
   - автоматическое получение перечня мер ЗИ из базового набора, согласно вышеопределенному уровню;
   - адаптация перечня мер в соответствии с особенностями объекта оценки: исключение мер, исходя из свойств системы, в соответствии с неактуальностью угрозы, а так же определение мер, заменяемых компенсирующими.

Анализ мер для реализации

1 этап – Аудит системы по требованиям НПА БР и серии ГОСТов.

Определение уровня защищенности выполняется не по ГОСТ Р 57580.1, а по НПА БР – положениям, требующим соответствия его критериям:

- БР 683-П от 17.04.2019 «Об установлении обязательных для кредитных организаций требований к обеспечению ЗИ при осуществлении банковской деятельности в целях противодействия осуществлению переводов денежных средств без согласия клиента»;
- БР 802-П от 25.07.2022 «О требованиях к ЗИ в платежной системе БР»;

- БР 821-П от 17.08.2023 «О требованиях к обеспечению ЗИ при осуществлении переводов денежных средств и о порядке осуществления БР контроля за соблюдением требований к обеспечению защиты информации при осуществлении переводов денежных средств».

Таблица 1

Определения уровня ЗИ

Минимальный (3й уровень ЗИ)	Стандартный (2й уровень ЗИ)	Усиленный (1й уровень ЗИ)
683-П от 17.04.2019		
не применяется	Кредитные организации (КО), не относящиеся к кредитным организациям, реализующим усиленный уровень ЗИ	Системно значимые КО, КО, выполняющие функции оператора услуг платежной инфраструктуры системно значимых платежных систем, КО, значимые на рынке платежных услуг
802-П от 25.07.2022		
не применяется	участники ССНП участники СБП ОУИО СБП	ОПКЦ СБП
821-П от 17.08.2023		
Банковские платежные агенты (субагенты)	ОУИО Операторы услуг платежной инфраструктуры, **не оказывающие** услуги платежной инфраструктуры в рамках системно значимых платежных систем	Операторы услуг платежной инфраструктуры, **оказывающие** услуги платежной инфраструктуры в рамках системно значимых платежных систем

Вспомогательными в определении уровня защиты, требуемого для реализации могут стать положения требований ГОСТ Р 57580.3-2022 и ГОСТ Р 57580.4-2022, согласно которым уровень защиты зависит не только от типа организации, но и от лицензии банка, а также от его финансовых показателей.

Для определения необходимого уровня защиты для этих ГОСТов Банк России выпустил Методические рекомендации № 7-МР «По управлению риском информационной безопасности и обеспечению операционной надежности» от 21.03.2024, в которых кредитным организациям рекомендовано обратиться к Положению Банка России от 08.04.2020 N 716-П «О требованиях к системе управления операционным риском в кредитной организации и банковской группе» от 8.04.2020, где сказано, что:

1. Усиленный уровень защиты ГОСТ Р 57580.3-2022 рекомендуется реализовывать банкам, если для них справедливо любое из условий: размер их активов на начало года составляет 500 млрд. руб. и более; имеется универсальная лицензия.
2. Стандартный уровень защиты ГОСТ Р 57580.3-2022 рекомендуется: банкам с базовой лицензией; небанковским кредитным организациям.
3. Усиленный уровень защиты ГОСТ Р 57580.4-2022 рекомендуется реализовывать банкам, если размер их активов на начало года составляет 500 млрд. руб. и более.
4. Стандартный уровень защиты ГОСТ Р 57580.4-2022 рекомендуется реализовывать всем остальным кредитным организациям, а именно: банкам с универсальной лицензией; банкам с базовой лицензией; небанковским кредитным организациям. [2].

1.1. Аудит по определению контуров защиты

В случае если в область оценки соответствия ЗИ входят несколько контуров безопасности с различными уровнями ЗИ, сформированными финансовой организацией в соответствии с требованиями ГОСТ Р 57580.1, для реализации требований к обеспечению ЗИ, установленных НПА БР, числовое значение оценки каждого процесса системы ЗИ E_i вычисляют:

- отдельно по контурам безопасности с одинаковым уровнем ЗИ;
- как сумму числовых значений оценок E_i для контура (контуров) безопасности с учетом их весовых коэффициентов [3].

Если внутри области оценки контуры безопасности имеют одинаковый уровень ЗИ, тогда пользователь оценивает их по отдельности. Если уровни разные, то выполняется оценка каждого контура по отдельности, а потом производится итоговый расчет.

При старте работы с программой пользователю предлагается выполнить выбор количества контуров безопасности и оценить каждый из них отдельно. А уже потом, в случае если имеются контуры с разными уровнями защищенности, итоговая E_i будет подсчитана для контуров безопасности с учетом их весовых коэффициентов.

1.2. Аудит по определению требуемого уровня ЗИ оцениваемого контура

Программа должна выводить диалоговое окно (Стартовое окно), в котором будет условие «Выберите требуемый уровень защиты информации», предполагающее следующие варианты ответов: уровень 3 – минимальный; уровень 2 – стандартный; уровень 1 – усиленный; затрудняюсь ответить.

В случае если пользователь не знает требуемый организации уровень защищенности – он выбирает вариант «затрудняюсь ответить». По этому действию он переходит на форму (вкладку) со вспомогательной информацией (Справочное окно), где должна быть представлена табл. 1 (Таблица определения уровня) и уточняющие подсказки из ГОСТов 57580.3 (и .4), которые помогут пользователю сделать выбор.

В случае выбора одного из 3х первых вариантов необходимо, чтобы программа осуществляла переход на следующую форму (вкладку), где должен быть представлен список необходимых мер (окно Выбор меры). Перечень мер защиты берется для каждого из 8 процессов в соответствии с ГОСТ Р 57580.1, в соответствии с выбранным уровнем (3,2,1) защищенности.

Формирование перечня мер для оценки

2.1. Автоматическая загрузка перечня базовых мер из ГОСТ Р 57580.1, соответствующих выбранному уровню защищенности

Согласно ГОСТ Р 57580.1 каждому уровню защищенности соответствуют определенный (фиксированный) минимальный набор мер из базового набора и требования по их реализации [4]. Обязательность реализации меры и способ ее реализации зависят от защищаемой информации: чем большая степень защиты, тем большее количество мер применяется и тем больше из них реализуются техни-

чески. Поэтому при автоматической загрузке перечня мер для каждого уровня защищенности исключаются неприменимые к нему меры и остаются только те, которые реализуются организационно или технически.

В программе после перехода к окну Выбора меры из файла загружается весь перечень мер для данного уровня защиты и выводится на экран. Должна быть представлена следующая справочная информация по оцениваемой мере: обозначение меры; описание меры; требование к реализации меры:

- «О» – реализация путем применения организационной меры ЗИ;
- «Т» – реализация путем применения технической меры ЗИ.

2.2. Редактирование перечня мер по применимости к конкретной ИС

Пользователю должна быть предоставлена возможность редактирования базового набора мер из ГОСТ 57580.1. Адаптация должна производиться под конкретную (оцениваемую) информационную систему с учетом:

- модели угроз и нарушителей безопасности информации;
- структурно-функциональных характеристик объектов информатизации;
- требований к ЗИ, установленных НПА [обзор ГОСТ Р 57580.1-2017].

Сам процесс адаптации включает в себя:

- исключение мер, исходя из особенностей системы, например, не связанных с используемыми информационными технологиями;
- исключение мер, исходя из особенностей системы, например, в соответствии с неактуальностью угрозы;
- определение мер, которые будут заменены компенсирующими (техническая реализация которых невозможна / нецелесообразна, например, по экономическим причинам).

В программе процесс адаптации должен быть реализован в окне Выбора мер следующим образом: напротив каждой меры из автоматически выгруженного списка должна быть возможность установ-

ки флага («+»), обозначающая выбор меры из базового набора / установки флага («к»), обозначающая применение компенсирующей меры / установки флага («–»), обозначающая отказ от выбора меры. По умолчанию везде стоит флаг «–».

Если мера не исключается, а заменяется компенсирующей (в поле стоит буква «к»), то при определении оценок для соответствующих процессов (подпроцессов) системы ЗИ и направлений ЗИ осуществляют оценку компенсирующих мер по такому же алгоритму, как если бы мера была базовой. Отличие состоит в том, что для применения компенсирующей меры должно быть обоснование, которое даже включается в отчет по результатам оценки соответствия ЗИ проверяющими организациями.

После того как пользователь примет решение по применимости каждой меры (все поля (флаги) установлены в одно из 3х возможных значений («+», «к», «–»)), он нажимает кнопку «сформировать адаптированный набор мер». По этой кнопке должно производиться действие (отрисовка новой формы – Адаптированный набор мер), где перечень мер будет уменьшен и видоизменен.

Из формы Адаптированный набор мер должна быть возможность вернуться на предыдущую форму – Выбор меры, где должны быть сохранены все значения, которые были установлены пользователем. Должна быть возможность изменить ранее установленные значения (внести правки, отредактировать) и по кнопке «сформировать адаптированный набор мер» опять попасть на форму Адаптированный набор мер, с учетом изменений.

После нескольких итераций пользователем будет сформирован адаптированный набор мер, которые подлежат оценке. Должна быть предусмотрена возможность сохранения этой информации в файл.

Заключение. Проблема определения набора мер ЗИ, подлежащих оценке, в соответствии с уровнем (ями) защиты для контура (ов) безопасности может быть решена с помощью анализа входной информации и поэтапного формировании адаптированного набора мер. Поскольку этот процесс является трудоемким и сложным, возникает потребность в его автоматизации, для чего в статье определены требования к технической реализации такого решения

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Яковлева А.О. и Жукова М.Н.* Разработка подхода к анализу соответствия результатов проведения аудита и применяемых оценочных процедур в рамках GAP анализа для кредитно-финансовой сферы // Вестник УрФО. Безопасность в информационной сфере» [Электронный ресурс]. – URL: https://info-secur.ru/index.php/ojs – в печати.
2. Появилась методология для реализации уровней защиты стандартов ГОСТ Р 57580.3-2022 и ГОСТ Р 57580.4-2022 [Электронный ресурс]. – URL: https://in4security.com/news/tpost/4h8kxg5kl1 (дата обращения: 05.05.2024).
3. ГОСТ Р 57580.2-2018 (дата введения: 28.03.2018).
4. ГОСТ Р 57580.1-2017 (дата введения: 08.08.2018).

UDC 004.056.55

Juan Ramirez

Mexico, Guadalajara, Operaciones Digitales y Procesamiento Integral de Datos Encriptados

ПРОГРАММИРОВАНИЕ СЛУЧАЙНОГО ИЗМЕНЕНИЯ ПЕРЕМЕННЫХ ДЛЯ ГОМОМОРФНОЕ ШИФРОВАНИЕ

Описана схема гомоморфного шифрования. Данные шифруются через случайное изменение переменных, а затем операции выполняются на зашифрованном тексте, который одновременно обрабатывается и расшифровать результат. Это улучшает традиционный гомоморфный Шифрование путем объединения двух этапов в один, Гарантия того, что данные могут быть использованы только по назначению. Шум и точность управляемы, без большой эффективности Компромиссы. Двухпартийная Схема между Клиентом и Банком была разработана в качестве Доказательства-Концепция. Банк не имеет доступа к входным данным, представляющим Клиентские данные, но способен обрабатывать зашифрованные векторы. Тем Метод является гибким и может быть адаптирован для разного числа сторон, управление разрешениями, уровни безопасности, оперативность требования и т.д.

Ключевые слова: *гомоморфный энкрип; доказательство с нулевым разглашением; безопасность данных; машинное обучение.*

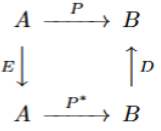
A homomorphic encryption scheme is described. Data is encrypted by randomly changing variables, and then operations are performed on the ciphertext, which is simultaneously processed and decrypted. It improves on traditional homomorphic encryption by combining two stages into one, ensuring that the data can only be used for its intended purpose. Noise and accuracy are manageable, without great efficiency tradeoffs. A two-party scheme between the Client and the Bank was developed as a Proof-of-Concept. The Bank does not have access to the input data representing the Client data, but is able to process the encrypted vectors. The method is flexible and can be adapted to different numbers of parties, permission management, security levels, responsiveness requirements, etc.

Keywords: *homomorphic encryption; zero-knowledge proof; data security; machine learning.*

I. Introduction

Traditionally, if one must process encrypted data, first the data is decrypted, and then the data is processed. This exposes the data to the second, and potentially third parties, in many instances. Homomorphic Encryption [1, 2] addresses this issue by changing the order of these two steps by first processing the data, while still encrypted, and then decrypting the result to plaintext. Machine Learning and AI applications that require sharing mass amounts of sensitive data, smart grids, large networks, traffic control, electronic voting, energy management, among many others, can be implemented if certain privacy issues are solved [3–7]. Let E a function that encrypts natural numbers. Suppose we encrypt two numbers x, y to obtain two new numbers Ex, Ey. We won't worry right now about which space the ciphertext is defined in, and suppose we have an operation $\oplus$ defined in that space. The operation of these gives $Ex \oplus Ey$. For most encryption functions, this is not equal to $E(x \oplus y)$. Therefore, decrypting $Ex \oplus Ey$ does not yield the expected result $x \oplus y$. If there exists a computable function D such that $x + y = D(Ex \oplus Ey)$, for any choice of $x, y \in \mathbb{N}$, we have a Partially Homomorphic Encryption Scheme for Addition (+). If the same property is also satisfied for multiplication ($\cdot$), then it is a Fully Homomorphic Encryption Scheme. The first difficulty in HE is that an operation and an encryption function are almost never homomorphic, $E(x + y) \neq Ex \oplus Ey$. Mathematical homomorphisms that can be used for HE are not practical solutions because of the complicated structures involved. Consequently, precision and noise are not easily mitigated [8, 9] making the algorithms unpractical or energy inefficient [10–12] in many scenarios. We explore a Keyless Decryption method that merges processing and decryption into a single step, avoiding Bootstrapping techniques. In this case, the processing function is the decryption key. The first party encrypts the inputs by applying a random change of variable from a library of encryption functions. The second party possesses a corresponding library of process/decryption functions and determines which of these functions will process/decrypt the result. Applying any other function of the library yields a meaningless answer. We describe an encrypted Credit Score calculation for a personal one-year loan as a Proof of Concept.

The commutative diagram for representing a HE Scheme, in its simplest conceptual form, is given by the equation $P = D \circ P^* \circ E$ shown below (1), where $P: A \to B$ is the process and P^* is the process applied to the encrypted data.



(1)

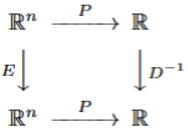
This diagram illustrates the basic relation for Encrypting the domain of the function and decrypting the result. Applying process P to the plaintext is equivalent to encrypting, then applying P^* and finally decrypting. We propose an HE scheme based on the concept of change of variable. Suppose a calculation $P: \mathbb{R}^n \to \mathbb{R}$ must be applied to an ordered n-tuple of variables $x = (x_i)_{i=1}^n = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$. We encrypt these n variables by applying a computable function $E: \mathbb{R}^n \to \mathbb{R}^m$ to the ordered n-tuple. An encryption function of P is a computable function $E: \mathbb{R}^n \to \mathbb{R}^m$ if there exists a computable function $P^*: \mathbb{R}^m \to \mathbb{R}$ such that $Px = (P^* \circ E)x$. In this case we say P, P^* are an ordered pair of homomorphic functions [13] with respect to encryption function E.

II. Encryption by change of variable

Suppose $P: \mathbb{R}^n \to \mathbb{R}$ is a function of n independent variables, and one party wishes for a second party to compute $P(x_1, x_2, \dots, x_n)$ without having the capability of knowing the input vector $(x_1, x_2, \dots, x_n)$. Let $E: \mathbb{R}^n \to \mathbb{R}^n$, defined by functions $e_1, e_2, \dots, e_n: \mathbb{R}^n \to \mathbb{R}$ such that

$$P \circ E = D^{-1} \circ P \tag{2}$$

for some invertible function $D^{-1}: \mathbb{R} \to \mathbb{R}$.

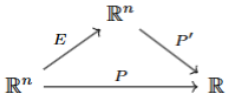


Suppose the change of variables E expresses $P \circ E$ in terms of an invertible function D^{-1}, of P. If the inverse function D is computable then we have encryption and decryption functions E, D, respectively. For example, let $P: \mathbb{R}^2 \to \mathbb{R}$ be the function $P(x,y) = x + y^2$, and let $e_1(x) = a^2x^2 + 2a^2xy^2$ and $e_2(y) = ay^2$, for some parameter $a \in \mathbb{R}$.

Then

$$(P \circ E)(x,y) = e_1(x) + \big(e_2(y)\big)^2 = a^2(x + y^2)^2 = a^2\big(P(x,y)\big)^2.$$

We have found $P \circ E$ to be expressed in terms of P, namely($P \circ$ ly$(P \circ E)(x,y) = a^2\big(P(x,y)\big)^2$. In this case, the function D^{-1} is given by$D^{-1}(x) = a^2x^2$. Therefore, the decryption function is given by $D(x) = \frac{\sqrt{x}}{a}$. This implies $P = D \circ P \circ E = \frac{1}{a}(P \circ E)^{\frac{1}{2}}$. Let us understand what this relation means, and what it permits. Suppose we wish for a second party to compute $P(x_0, y_0) = x_0 + y_0^2$, for constants $x_0, y_0 \in \mathbb{R}$, but we do not wish to share the plaintext vector (x_0, y_0) with the second party. First, we encrypt the inputs, using the function E described above, to obtain $E(x_0, y_0) = \big(e_1(x_0, y_0), e_2(x_0, y_0)\big)$, and send this encrypted vector to the second party. The second party will then take this encrypted vector and apply function $P' = D \circ P$ to obtain the desired result. We collapse (1) into a three diagram. If we consider the composition $P' = D \circ P$ as a single function, the resulting relation is $P = P' \circ E$.

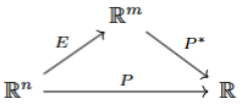


(3)

Can the second party decrypt the encrypted vector to find the original inputs x_0, y_0 with the information available? The information available is 1) Processing function P, 2) Encryption function E and Encrypted Variables e_1, e_2, and 3) Final result p_0. From 1) and 3) we have equation $x_0 + y_0^2 = p_0$, where $p_0 = (P' \circ E)(x_0, y_0)$. Two more equations are given by $a^2x_0^2 + 2a^2x_0y_0^2 = e_1$ and $ay_0^2 = e_2$. A solution for a, x_0, y_0, of this system of equations must be found. If we find a solution for a, we can then calculate x, y. Let us specify the bit size of the values in question. We have 64-bit inputs x_0, y_0. We also have a 128-bit encryption key, a. This key is

unknown to the second party, otherwise it could easily decrypt e_1, e_2 to find x_0, y_0. For purposes of finding solutions to this system of equations the key a is an unknown integer of 128 bits. Finding the original inputs is equivalent to finding the key a. If the second party knows the inner workings of the scheme, they would know to start by finding the factors of the 256-bit number e_2. Now that we understand where vulnerabilities arise, we can ask a few questions to maximize security. Let us start by asking if a type of integer is optimal for the security parameter? For example, if a is always chosen to be a prime number, then finding a is trivial. To find a, you must simply find the factors of $e_2 = ay^2$ that are smaller than 64-bits. Once we have the 128-bit key a, we can calculate y and then x. Thus, the key should not be chosen to be a prime number. The more prime factors in a, the better. For every encryption function E, of some processing function P, we have different security parameters. In each case, the keys have different required forms to maximize security.

We give a definition of encryption and decryption functions that supersedes the previous ones. If, more generally, given an encryption function $E: \mathbb{R}^n \to \mathbb{R}^m$ we can find a computable function $P^*: \mathbb{R}^m \to \mathbb{R}$ such that $P = P^* \circ E$, then we have an encryption scheme for process P.

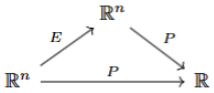


There is no straightforward method for finding the most secure and efficient encryption and processing functions E, P^*, for a given process P. We use the fact that certain encryption functions are safe (the keys, and therefore the plaintext inputs, cannot easily be found). Even if the encryption and processing functions E, P^* are known, finding the keys is a very difficult problem. One way to ensure the encryption function remains unknown is to have a large library of encryption functions for each processing function, and then randomly select an encryption function each time. This solution is discussed in Sections 6 and 7. The method proposed here allows for an FHE scheme that can be adapted for different number of parties, permission configurations and security needs. In Section 5, we outline a simple two-party scheme to simulate a personal loan Credit Score where the first party is a Client, and the second party is a Bank. The

Client will encrypt the inputs and share them with the Bank. The Bank receives these encrypted values, processes the encrypted data and sends the result back to the Client.

III. Keyless decryption

We will illus trate a special case of relation (3). Let $P(x,y,z)=\frac{x^2+y^2}{z}$. If we make the change of variables $e_1(x)=ax,\ e_2(y)=ay,\ e_3(z)=a^2z$, then we have $P=P\circ E$. Again, $P\circ E$ is a function of P. In this case $D^{-1}=I$ is the identity function. This example of $P(x,y,z)=\frac{(ax)^2+(ay)^2}{a^2z}$ cancels nicely with the encryption function, so that $D=I$ is the identity function. When this happens, we say this is a Keyless Decryption Scheme because $P'=I\circ P=P$. In this case, the function that must be applied to the encrypted data is the same function that would be applied to the original plaintext. No new function has be defined because, $P'=P$.



This example of Keyless Decryption is not semantically secure for several reasons, but non-trivial and semantically secure examples of Keyless Decryption can be found for certain functions. In this example, if any of x,y,z is zero, then its corresponding Encrypted Variable will be zero. Secondly, the bit-length of the Variables is easily deduced from the Encrypted Variables, which makes it useless for almost any application. Ultimately, the encryption function of this example does not work for the following reason. It is easy to find Key a, to calculate the Inputs. The Key is the number that divides e_1,e_2 once and divides e_3 two times. There are different techniques for defining secure encryption functions and keys without recurring to excessive bit-lengths, because there are encryption functions whose equations do not determine the keys easily. The solution presented in Sections 6 and 7 uses several strong encryption functions, and randomly chooses a different one each time, adding an extra layer of security. Given a large library of encryption functions $\{E_j\}_j$, for P, each encryption function E_j is associated to a unique processing function P_j^*.

V. Mathematical model

We define encryption and processing functions that yield a decrypted plaintext Credit Score, to be shared with the Client. The Credit Score function is normalized. If the income to expense ratio and the worth to debt ratio go to infinity, the Credit Score is 10. If, on the other hand, the expenses and debt grow with respect to income and worth, the Credit Score is 0. The minimum approval score is suggested at 5 points, given the current settings and weights of the parameters, but these can be modified for different cases. A reasonably calibrated model for loan approvals based on a client's financial data (income, capital, debt, expenses, etc.) is proposed. Parameters can be modified to adapt the Credit Score model to different scenarios (different types of loans such as micro loan, business loan, loan period, interest, etc.). The proposed processing function P is

$$P(NI, TE, W, TD) = \frac{A\,NI}{\sqrt{NI^2 + \alpha\,TE^2}} + \frac{B\,W}{\sqrt{W^2 + \beta\,TD^2}},$$

where A, α, B, β are parameters, which we will set to numeric values to calibrate our model. There are other parameters such as the interest rate and loan period which will be set to 15% interest rate for a one-year loan, and which are used to calculate the Newly Incurred Debt, Service to Debt, Total Debt and Total Expenses (including the new loan).

VI. Randomizing methods

We can add another layer of security. Suppose we have a library of k different encryption functions $\{E_j\}_{j=1}^{k}$, of a given processing function P. If the client encrypts the variables using E_i, for some $E_i \in \{E_j\}_j$, then the encrypted variables will have to be processed with the corresponding processing function P_j^*. Furthermore, the codimension of E_j can be different from another encryption functions codimension. It is possible to implement a randomized selection of the encryption functions that adds an extra layer of security. After transforming the inputs into the variables, in Module I, the variables are sent to Module II. Now, when Module II receives the variables, it chooses a method from the library of encryption functions $\{E_j\}_j$, to encrypt the variables. Once j is fixed, the vector of variables in

$\mathbb{R}^n$ will be transformed into a new vector of encrypted variables. The bank has a corresponding library of processing functions $\{P_j^*\}_j$. Upon receiving the vector of encrypted variables, the bank will be able to identify and run the correct processing function. Having a large library of encryption methods is beneficial for the security of the overall scheme because the data type and the number of encrypted variables can be different each time process P has to be applied to a vector x, making it harder to implement pattern recognition. A basic version of the personal loan Credit Score, detailed above, has been implemented and will be made available for auditing purposes, as well as for establishing collaboration opportunities.

VII. Generalization

In many instances it is desirable to have data encrypted in such a way that it can then be re-encrypted for use in one of many different processing functions. Consider a one-party scheme where the User has a numerical database $\mathcal{E}(DB)$ encrypted at rest and wishes to perform operations on elements of the database but wants to maintain the data confidential. The finite library of functions for processing the data is $\{P\}_P = \{R, S, \dots, T\}$. The User requests for any of the functions to be applied to a vector of the database. Suppose we have a library of at least two processing functions, where $R(x, y) = x + y$ and $S(x, y) = xy$. Every processing function $P \in \{P\}_P$ has its own library of encryption functions. The encryption libraries are $\{E_{R,i}\}_i = \{E_{R,1}, E_{R,2}, \dots\}, \{E_{S,j}\}_j = \{E_{S,1}, E_{S,2}, \dots\}, \dots$. Each encryption library has its own cardinality. The User sends a request to the database which includes an encrypted index for a processing function $P: \mathbb{R}^n \to \mathbb{R}$, and the encrypted addresses of the elements of the database. The request is a vector of $n + 1$ coordinates $(\#P, x_1, x_2, \dots, x_n)$. The first coordinate of the request is an index, $\#P$, indicating to the database the processing function. The other n coordinates of the request are the addresses of the elements of the database. Then, the database randomly selects an encryption function $E: \mathbb{R}^n \to \mathbb{R}^m$ from the set $\{E_{P,j}\}_j$. The database has a library of processing tions$\{P^*_E\}_{P,E} = \{R^*_{E_{R,1}}, R^*_{E_{R,2}}, \dots, S^*_{E_{S,1}}, S^*_{E_{S,2}}, \dots\}$, for all $P \in \{P_j\}_j$ and all $E \in \{E_{P,j}\}_j$. The processing function P^*_E depends on the processing func-

tion P and the encryption function E, so that if we change the process P then the encrypted processing function P_E^* is also different. Similarly, if we change the encryption function E then the function P_E^* changes again. The database identifies and applies the correct function P_E^* to the other n encrypted values.

Conclusions

Although HE is a promising concept in the science of cryptography, the same reasons that make it safe also make it non applicable in many situations. Current energy and time efficiency standards of HE are not met in a wide range of crucial applications. This change of variable method offers many advantages including the fact that the encrypted data can only be used for the intended purposes, and noise and precision are managed without significant efficiency trade-offs. This scheme is applicable to a wide range of processing functions, including addition and multiplication from which a FHE scheme can be constructed. Trigonometric functions and the numeric derivative can also be homomorphically encrypted. It is a flexible scheme that can be adapted to different number of participants, permission configurations, security demands, efficiency requirements, etc. These characteristics make it a far and wide reaching solution with applications in online security, private Digital Signal Processing, smart cities and traffic problems, Machine Learning such as in specific cases of AI and Neural Network training utilizing sensitive data, operating vectors of encrypted databases, smart grids and energy management, among other activities directly dependent on the difficult marriage of data security and efficient computing. The proposed method can be integrated to SoC by implementing this encryption scheme in a processor architecture with a Simple and Linear Fast Adder having a linear and scalable topology [14–16]. Encrypted Processing Units can help achieve secure cloud computing without sacrificing efficiency at hardware and software level.

BIBLIOGRAPHICAL LIST

1. *Ronald L. Rivest, Len Adleman, and Michael L. Detouzos.* On data banks and privacy homomorphisms // In Foundations of Secure Computation. – P. 165-179. – Academic Press, 1978. Available at https://people.csail.mit.edu/rivest/pubs.html#RAD78.

2. *Gentry C.* A Fully Homomorphic Encryption Scheme: Doctoral Dissertation, Symposium on the Theory of Computing, NY, New York, USA, 2009.
3. *Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing*. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy // In 33rd International Conference on Machine Learning (ICML 2016), volume 48 of Proceedings of Machine Learning Research. – PMLR, 2016. – P 201-210. – URL: http://proceedings.mlr.press/v48/giladbachrach16.html.
4. *Florian Bourse, Michele Minelli, Matthias Minihold, and Pascal Paillier*. Fast Homomorphic Evaluation of Deep Discretized Neural Networks // In Advances in Cryptology – CRYPTO 2018. Part III. Vol. 10993 of Lecture Notes in Computer Science. – P 483-512. – Springer, 2018. – DOI: 10.1007/978-3-319-96878-0 17.
5. *Christina Boura, Nicolas Gama, Mariya Georgieva, and Dimitar Jetchev*. Simulating Homomorphic Evaluation of Deep Learning Predictions // In Cyber Security Cryptography and Machine Learning (CSCML 2019). Vol. 11527 of Lecture Notes in Computer Science. – Springer, 2019. – P. 212-230. – DOI: 10.1007/ 978-3-030-20951-3 20.
6. *Marcelo Blatt, Alexander Gusev, Yuriy Polyakov, and Shafi Goldwasser.* Secure Large-Scale Genome-Wide Association Studies Using Homomorphic Encryption // Cryptology ePrint Archive, Report 2020/563. – 2020. – https://ia.cr/2020/563.
7. iDASH secure genome analysis competition. http://www.humangenomeprivacy.org.
8. *Martin R. Albrecht, Rachel Player, and Sam Scott.* On the concrete hardness of learning with errors // Journal of Mathematical Cryptology. – 2015. – 9 (3). – P. 169-203. – DOI: 10.1515/jmc-2015-0016.
9. *Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé*. Classical hardness of learning with errors // In 45th Annual ACM Symposium on Theory of Computing. – ACM Press, 2013. – P. 575-584. – DOI: 10.1145/2488608.2488680.
10. *Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachéne.* TFHE: Fast fully homomorphic encryption over the torus // Journal of Cryptology. Earlier versions in ASIACRYPT 2016 and 2017. – 2020. – 33 (1). – P. 34-91. – DOI: 10.1007/s00145-019-09319-x.
11. *Léo Ducas and Daniele Micciancio.* FHEW: Bootstrapping Homomorphic Encryption in Less than a Second // In Advances in Cryptology-EUROCRYPT 2015. Part I. Volume 9056 of Lecture Notes in Computer Science. – Springer, 2015. – P. 617-640. – DOI: 10.1007/978-3-662-46800-5 24.

12. *Joppe W. Bos, Kristin E. Lauter, et. al.* Improved security for a ring-based fully homomorphic encryption scheme // In Cryptography and Coding (IMACC 2013). Vol. 8308 of Lecture Notes in Computer Science. – Springer, 2013. – P. 45-64. – DOI: 10.1007/978-3-642-45239-0 4.
13. *Ramirez J.* Systems and Categories // arXiv:1509.03649v5 [math.CT]. – 2015.
14. *Ramirez J. Simple* and Linear Fast Adder of Multiple Inputs and Its Implementation in a Compute-In-Memory Architecture // 2024 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA), Victoria, Seychelles, 2024. – P. 1-11. – DOI: 10.1109/ACDSA59508.2024.10467957.
15. *Ramirez J.* SIMPLE AND LINEAR FAST ADDER // WIPO, Patentscope. Publication Number: WO/2023/220537. Publication Date: 16/11/2023. Applicant's name: Juan Pablo Ramirez.
16. *Ramirez J.* Canonical Set Theory with Applications from Matrix Operations and Data Structures to Homomorphic Encryption. –2023. Author's personal homepage: www.binaryprojx.

UDC 004.056

K.T. Algazy, K.S. Sakan, N.A. Kapalova

Kazakhstan, Almaty, Institute of Information and Computing Technologies

POST-QUANTUM CRYPTOGRAPHY BASED ON HASH FUNCTIONS

The advent of quantum computing poses a threat to the security of traditional cryptographic systems, making the development of post-quantum cryptographic algorithms a particularly urgent task. This article provides an overview of existing digital signature schemes resistant to quantum computer attacks, with a special focus on methods based on hash functions. The review covers the main concepts underlying post-quantum hash-based signatures and their implementation methods. The article discusses both classical schemes, such as the Lamport-Diffie and Merkle schemes, as well as more recent developments, including improved versions with optimized performance and smaller key sizes.

Keywords*: post-quantum cryptography; hash function; digital signature; key; security.*

The concept of designing an electronic digital signature based on hash functions emerged in 1979 in the work of the American scientist Lamport. When creating a signature using a pseudo-random number generator (PRNG), 256 pairs of random 256-bit numbers $(a_i, b_i), i = \overline{0{,}255}$, are generated, which are considered the secret key of the system. By hashing the pairs of numbers (a_i, b_i), corresponding pairs of public keys $(a_i', b_i'), i = \overline{0{,}255}$, are created. In general, in this scheme, any cryptographically secure one-way function H can also be applied [1].

$$\begin{pmatrix} a_0 & b_0 \\ \dots & \dots \\ a_{255} & b_{255} \end{pmatrix} \overset{H}{\rightarrow} \begin{pmatrix} a_0' & b_0' \\ \dots & \dots \\ b_{255}' & b_{255}' \end{pmatrix}.$$

To sign a message, first, its hash code, with a length of 256 bits, is calculated. Each of the bits of the resulting hash code is used to select a specific number from the next pair of secret keys in sequence. The signature is created by selecting the first (left) number from the pair of secret

keys (a_i, b_i), if the current value of the hash code bit is 0, and the second (right) number from the same pair if the value is 1. As a result, the total length of the signature is 256 * 256 bits = 65536 bits or 8 kB.

To verify that the obtained signature is correct, the verifier hashes the received message and the signature elements. To select the corresponding value from each pair of public keys (a_i', b_i'), it uses the bits of the hash code. That is, if the value of the bit is 0, the verifier takes the first number from the pair (a_i', b_i'), and if it is 1, the verifier takes the second number from the pair (a_i', b_i'). The resulting bit value is compared to the corresponding bit of the signature hash code, and if these values are equal, the process moves on to check the next value. Ultimately, the signature is considered valid if all 256 comparison operations yield a positive result [2].

One of the well-known one-time signature schemes (OTS), used in many modern post-quantum signatures, is the Winternitz OTS (WOTS) scheme [3]. In the construction of a signature in the WOTS scheme, a sequence of secret keys $SK = (SK_0, SK_1, \dots, SK_{m/w})$ is first generated using a random number generator (RNG). Then, a sequence of public keys $PK = (PK_0, PK_1, \dots, PK_{m/w})$ is generated using the hash function H applied $2^w - 1$ times (Fig. 1).

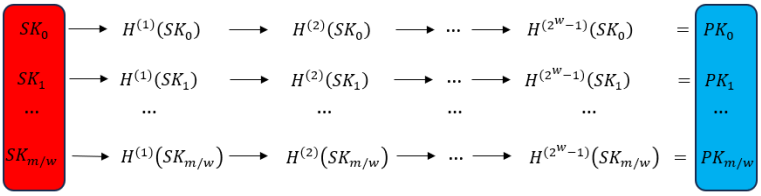


Fig. 1. The process of obtaining a public key in the WOTS scheme

To create a digital signature, the message M of length m bits is divided into blocks of length w, i.e., $M = (M_0, M_1, ,.., M_{m/w})$. Then, from the numerical representations of each block M_i, a sequence $(a_0, a_1, .., a_{m/w})$ is formed. Using this sequence, the signature σ of the message M is computed as follows: $\sigma = (\sigma_0, \sigma_1, \dots, \sigma_{m/w}) = \left(H^{(a_0)}(SK_0), H^{(a_1)}(SK_1), \dots, H^{(a_{m/w})}(SK_{m/w})\right)$.

The WOTS scheme has several variations. One of them is the WOTS+ scheme, incorporated into the structure of the SPHINCS+ algorithm, which was presented in the NIST competition and selected after three rounds. In this scheme, after splitting the transmitted message M into blocks, a checksum $C = \sum_{i=1}^{l_1}(2^w - 1 - M_i)$ is added to its end. Let's denote the total number of blocks as l, consisting of the sum of two numbers $l = l_1 + l_2$, defined as follows: $l_1 = \lceil n/\log(w) \rceil$, $l_2 = \left\lfloor \frac{log(l_1(w-1))}{\log(w)} \right\rfloor + 1$, where n is the number of bits in the public or secret keys.

Thus, the blocks $(M_1, \dots, M_{l_1},\ C_1, .., C_{l_2})$ will be formed, and from the numerical values of each of these blocks M_i and C_i, a sequence $(a_0, a_1, .., a_{l-1})$ is generated. Using this sequence, the signature σ is calculated as follows:

$$\sigma = (\sigma_0, \sigma_1, \dots, \sigma_{l-1}) = \left(H^{(a_0)}(SK_0), H^{(a_1)}(SK_1), \dots, H^{(a_{l-1})}(SK_{l-1})\right).$$

During signature verification, all computational actions are performed similarly, and additionally, at the last stage, B is calculated as follows:

$$B = (b_0, b_1, \dots, b_{l-1}) = \left(H^{(2^w - a_0)}(\sigma_0), H^{(2^w - a_1)}(\sigma_1), \dots, H^{(2^w - a_{l-1})}(\sigma_{l-1})\right).$$

Ultimately, if the identity $B = PK$ holds true, the signature is considered authentic; otherwise, it is not.

Another key distinction of the WOTS+ scheme is that to enhance its cryptographic resistance, it employs a public key consisting of $2^l - 1$ numbers, denoted as $PK_t = \left(k_0, k_1, \dots, k_{2^l - 1}\right)$, and additionally utilizes a chain function $c^i(x, PK_t)$.

The function $c^i(x, PK_t)$ takes a new parameter $x \in \{0,1\}^n$ and is defined by the following formula:

$$c^i(x, PK_t) = \begin{cases} x, \text{if } i = 0 \\ H\left(c^i(x, PK_t) \oplus k^i\right), \text{если } i > 0. \end{cases}$$

The advantage of using a chain function in the WOTS+ scheme is that the high collision resistance of the hash function used in it is not a mandatory condition.

Nevertheless, since the mentioned schemes belong to the group of one-time signatures, it is necessary to carefully manage the keys when generating signatures for multiple messages. Therefore, in addition to them, schemes with few-time signatures (FTS) are also being developed. Among the most popular of these schemes is the HORS (Hash to Obtain Random Subset) scheme [4].

To generate a signature according to the HORS scheme, the message to be signed, with a length of l, is evenly divided into k parts. A pseudorandom number generator (PRNG) forms the secret key $SK = (SK_0, SK_1, \dots, SK_{t-1})$, consisting of $t = 2^l$ key elements. Each element $SK_i, i = \overline{0, t-1}$, determines the public key $PK = (PK_0, PK_1, \dots, PK_{t-1})$ as the output of the hash function. To obtain the signature of the message, the numerical value of all k частей parts is calculated, and their indices are selected from the set of secret keys based on these values: $\sigma = (\sigma_0, \sigma_1, \dots, \sigma_{k-1}) = \left(SK_{n_0}, SK_{n_1}, \dots, SK_{n_{k-1}}\right)$. Here, n_0 corresponds to the first, n_1 to the second, and so on, and n_{k-1} represents the kth numerical values.

Several modifications of the HORS scheme have also been developed, such as HORSIC and HORSIC+. HORSIC employs a bijective function to transform all k parts into numerical values. Besides, a method for reducing the signature size is considered. In HORSIC+, the same chaining function as in the WOTS+ scheme is used to enhance security, but with the consideration of being resistant to chosen message attacks [5].

The drawback of the aforementioned schemes is that in some, a key cannot be used more than once, while in others, the level of security continues to decrease upon reusing the same key. To address these limitations, it is recommended to use a Merkle tree (binary hash tree). When constructing a binary tree, a single overall hash value, called the Merkle root, covering all data parts is computed. By using a single public key and a Merkle tree of height n, it is possible to create a common signature for 2^n messages [6, 7]. To implement a binary hash tree, pairs of 2^n secret and public keys are placed in the leaf nodes of the tree. Each parent node is determined using the formula $t_{i,j} = H\left(t_{i-1,2j} || t_{i-1,2j+1}\right)$.

To create a signature for the current message, a random key that has not been used before is selected. Then, using this key and one of the signature schemes (Lamport, WOTS, etc.), a signature σ is formed. Subse-

quently, the sender transmits $(M, \sigma, auth_0, auth_1, \ldots, auth_{n-1})$, i.e., the message, its signature, and the authentication path. Fig. 2 illustrates an example of a Merkle tree with a height of 3 when selecting the 5th key.

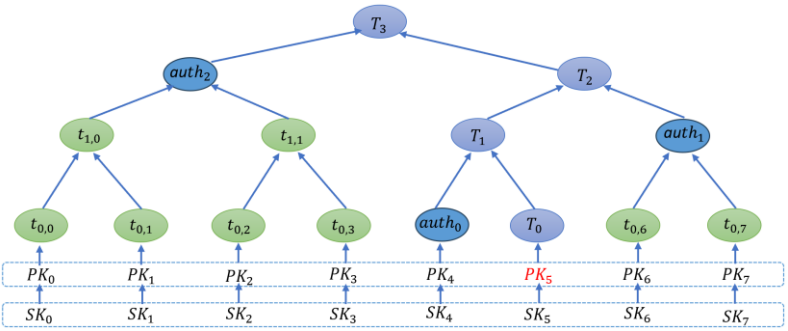


Fig. 2. Signature using a Merkle tree

To verify the signed message M, first, its signature σ is checked. If it's correct, then by hashing the selected *i*th public key, T_0 is computed, i.e., $T_0 = H(PK_i)$. Other nodes T_i of the tree are computed as $H(T_{i-1}||auth_{i-1})$. If the value of T_n equals the public key, then the signature is considered valid.

Using the Merkle tree to manage keys of the WOTS+ scheme, a new scheme called XMSS (eXtended Merkle Signature Scheme) was created [8]. In XMSS, the root of the Merkle tree is taken as the public key, and each leaf is the Merkle root from the set of WOTS+ public keys. Such a tree is also called an "L-tree". To sign a message M, a leaf of the tree that has not been used before is selected, and it uses secret keys to generate a WOTS+ signature.

It is necessary to store in memory all secret keys that have been used in signatures formed using one or more Merkle trees, and these keys should not be reused. Therefore, such signatures are called stateful signatures. Randomly reusing any key in the system can compromise the entire system. However, such signatures also have their advantages, including their fast generation and relatively short size.

Tracking the pair of used keys in one-time signatures is considered one of the main drawbacks of stateful schemes. To address this issue, schemes have emerged that allow for stateless operation. Unlike OTS

schemes, stateless signature schemes can be used multiple times. For example, one such scheme is the HORS-T signature scheme (HORS with a tree). This digital signature scheme eliminates the need to store the secret key for each signature generation.

In addition to the schemes mentioned above, there are several other ways to create HBS. Among them, it is worth noting the SPHINCS+ digital signature scheme, which was selected as a result of the NIST competition. In this scheme, there is also no need to track the state. One of the early versions of the SPHINCS+ algorithm, the SPHINCS algorithm, was developed using HORS-T and WOTS+ schemes, and to compute the root of the Merkle tree, keys need to be pre-generated. In the SPHINCS algorithm, by using a hypertree and a random string key address scheme, it was possible to manage multiple keys without computing all the leaves of the tree.

The SPHINCS+ algorithm, when signing a message, uses an enhanced version of HORS called the Forest of Random Subsets (FORS) scheme and a tunable hash function for security. In FORS, k binary trees of height n are considered. Since each tree contains $t = 2^n$ secret keys, the total number of keys will be $k \times t$. The roots of these trees are hashed together to form the FORS root.

The SPHINCS+ algorithm also utilizes the WOTS+ one-time signature scheme. Here, WOTS+ signs not the message itself, but the roots of the FORS trees using WOTS+ secret keys. Accordingly, the WOTS+ public keys form the leaves in d layers of the hyper-tree. Then, the tree roots in the bottom layer of the Merkle tree are signed via WOTS+, and the corresponding WOTS+ public keys form the leaves of the subtree in the upper layer. Thus, the subtree root is signed sequentially with its corresponding subtree layers until it reaches the upper subtree (Fig. 3). The top-level subtree forms the root of the hyper-tree, which the verifier trusts [9].

The SPHINCS+ signature consists of the FORS signature, WOTS+ signatures at each layer, and authentication paths leading to the roots of the subtrees, signed by the hyper-tree root. It can be represented as $\Sigma = \left(M, \sigma_F, \sigma_0,\ auth_{A_0}, \sigma_1,\ auth_{A_1}, \dots, \sigma_{d-1},\ auth_{A_{d-1}}\right)$.

In this section, brief descriptions of signature schemes Lamport, WOTS, WOTS+, HORS, and some others are provided. In general, algorithms based on post-quantum hash functions can be divided into two groups: stateful and stateless. Fig. 4 shows schemes developed to date, grouped by structures [10, 11].

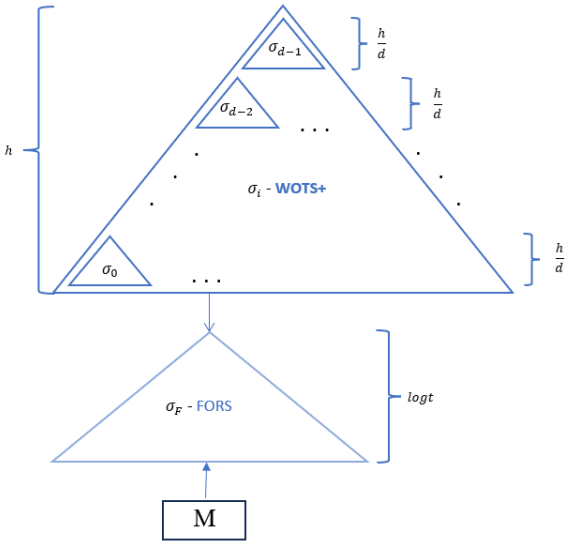


Fig. 3. Structure of the SPHINCS+ algorithm

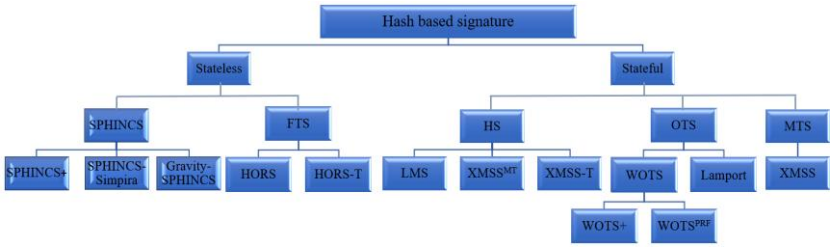


Fig. 4. Classification of hash-based digital signatures

There is every reason to believe that hash-based cryptography will remain one of the priority directions in post-quantum cryptography. Algorithms in this field will rely solely on cryptographically secure hash functions for generating and verifying digital signatures.

REFERENCES

1. *Li S., Chen Y., Chen L., Liao J., Kuang C., Li K., Liang W., Xiong N.* Post-Quantum Security: Opportunities and Challenges // Sensors. – 2023. – 23. – 8744. – https://doi.org/10.3390/s23218744.

2. *Moldovyan D.N., Moldovyan A.A., Moldovyan N.A*. Post-quantum signature schemes for efficient hardware implementation // Microprocessors and Microsystems. – 2021. – Vol. 80. – 103487. – https://doi.org/10.1016/ j.micpro.2020.103487.
3. *Kumar M.* Post-quantum cryptography Algorithm's standardization and performance analysis // Array. – 2022. – Vol. 15, 100242. – https://doi.org/10.1016/ j.array.2022.100242.
4. *Buchmann J., Lauter K., Mosca M.* Postquantum Cryptography – State of the Art // in IEEE Security & Privacy. – 2017. – Vol. 15, No. 4. – P. 12-13. – DOI: 10.1109/MSP.2017.3151326.
5. *Lee J., Park Y.* HORSIC+: An Efficient Post-Quantum Few-Time Signature Scheme // Applied Sciences. – 2021. – 11(16):7350. – https://doi.org/10.3390/ app11167350.
6. *Shahid F., Khan A., Malik S.R., Choo K.R.* WOTS-S: A Quantum Secure Compact Signature Scheme for Distributed Ledger // Information Sciences. – 2020. – Vol. 539. – P. 229-249. – https://doi.org/10.1016/j.ins.2020.05.024.
7. *Cavaliere F., Mattsson J., Smeets B.* The security implications of quantum cryptography and quantum computing // Network Security. – 2020. – Vol. 2020, Issue 9. – P. 9-15. – https://doi.org/10.1016/S1353-4858(20)30105-7.
8. *Hülsing A., Rausch L., Buchmann J*. Optimal Parameters for XMSSMT / In: Cuzzocrea, A., Kittl, C., Simos, D.E., Weippl, E., Xu, L. (eds) // Security Engineering and Intelligence Informatics. CD-ARES 2013. Lecture Notes in Computer Science. Vol. 8128. – Springer, Berlin, Heidelberg, 2013. – P. 194-208. – https://doi.org/10.1007/978-3-642-40588-4_14.
9. *Bernstein D.J., Hülsing A., Kolbl S., Niederhagen R., Rijneveld J., Schwabe P.* The SPHINCS + signature framework // In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19). – 2019, 2129–46. – https://doi.org/10.1145/3319535.3363229.
10. Contribution to The Handbook of Information. Available online: https://blkcipher.pl/assets/pdfs/NPDF-32.pdf (accessed on 6 January 2024).
11. *Nejatollahi H., Dutt N., Ray S., Regazzoni F., Banerjee I., Cammarota R.* Post-quantum lattice-based cryptography implementations // ACM Comput. Surv. – 2022. – 51, 129. – P. 1-41. – https://doi.org/10.1145/3292548.

СОДЕРЖАНИЕ

Научное издание

СОВРЕМЕННЫЕ МЕТОДЫ, СРЕДСТВА И ТЕХНОЛОГИИ ЗАЩИТЫ ИНФОРМАЦИИ – 2024

Сборник трудов
XV Международной научно-практической
конференции имени Олега Борисовича Макаревича

Таганрог, 11–15 сентября 2024

Ответственная за выпуск Е.А. Ищукова
Компьютерная верстка Н.В. Ярошевич

Электронное издание

Подписано к использованию 19.12.2024. Заказ № 9781. Тираж 10 экз.
Усл. печ. л. 15,2. Уч.-изд. л. 11,9.
Издательство Южного федерального университета
Отдел полиграфической, корпоративной и сувенирной продукции
Издательско-полиграфического комплекса КИБИ МЕДИА ЦЕНТРА ЮФУ
344090, г. Ростов-на-Дону, пр-т Стачки, 200/1, тел. (863) 243-41-66