УДК 004.056 Вестник УрФО № 1(59) / 2026, с. 28–35

Зуев А.Д., Зюляркина Н.Д., Соколов А.Н.

DOI: 10.14529/secur260103

ПОВЫШЕНИЕ ЗАЩИЩЕННОСТИ API-КЛЮЧЕЙ В РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ С ИСПОЛЬЗОВАНИЕМ СХЕМЫ РАЗДЕЛЕНИЯ СЕКРЕТА ШАМИРА

В статье рассматривается проблема обеспечения конфиденциальности API-ключей от банковских сервисов при интеграции финансовых сервисов в CRM-системы. Проанализирована архитектура распределенной информационной системы хранения ключей и выявлена уязвимость, связанная с централизованным хранением ключей шифрования на сервере-шифровальщике. Предложен метод повышения устойчивости системы к компрометации отдельных узлов посредством внедрения схемы разделения секрета Шамира. В отличие от существующих решений, использующих аппаратные модули безопасности (HSM) или единый сервер ключей, предложенная модель обеспечивает защиту данных даже при полной компрометации одного из узлов инфраструктуры. Описана математическая модель распределения ключей над конечными полями и архитектура новой информационной системы. Результаты апробированы на демонстрационном стенде с использованием сертифицированных средств защиты информации и отечественного программного обеспечения (ОС Astra Linux, шифр «Кузнечик»). Проведены замеры производительности с применением статистических методов оценки, показавшие приемлемые временные затраты на операции разделения и восстановления ключей.

Ключевые слова: *информационная безопасность, API-ключи, схема Шамира, разделение секрета, распределенные системы, шифр «Кузнечик», Astra Linux, криптографическая защита, производительность, доверительные интервалы.*

SECURING STORAGE OF API-KEYS IN DISTRIBUTED INFORMATION SYSTEMS USING SHAMIR'S SECRET SHARING SCHEMA

This paper investigates the problem of ensuring confidentiality of API-keys for external financial services during integration into CRM-platforms. A distributed API-key storage information system was analyzed and a vulnerability in centralized API-key storage was found. It is proposed to implement Shamir's secret sharing schema to resist data leaks from individual nodes in an information system. Unlike existing solutions using hardware security modules (HSM) or centralized key storage, the proposed model guarantees confidentiality even if an individual server in the architecture is fully compromised. A mathematical model for a secret sharing schema over finite fields and an improved information system structure are described. Results were obtained through testing on a demonstration testbed employing certified information security tools and domestic software (Astra Linux OS, "Kuznechik" encryption). Performance measurements were conducted using statistical evaluation methods, demonstrating acceptable time costs for key splitting and recovery operations.

Keywords: *information security, API-keys, Shamir's schema, secret sharing, distributed systems, "Kuznechik" encryption, Astra Linux, cryptographic security, performance, trust intervals.*

При цифровизации финансовых инструментов одной из важных задач является безопасная интеграция сторонних сервисов через программные интерфейсы (API). Вопросы защиты информации требуют решения при работе с API-ключами банковских сервисов, доступ к которым позволяет инициировать финансовые транзакции и получать конфиденциальные данные клиентов. Обычно на практике такие ключи хранятся в зашифрованном виде в базах данных информационных систем (ИС). При этом безопасность ИС в этом случае сводится к безопасности сервера, хранящего мастер-ключи для расшифровки [1].

Компрометация такого сервера ведет к полной утечке секретных данных. Существующие решения, такие как аппаратные модули безопасности (HSM), обеспечивают высокий уровень защиты, но характеризуются высокой стоимостью и сложностью интеграции в веб-ориентированные архитектуры на базе стека LAMP/LEMP. В связи с этим актуальной является разработка программных методов распределенного хранения ключей, обеспечивающих конфиденциальность даже при частичном перехвате контроля над инфраструктурой.

Целью данной работы является разработка архитектуры информационной системы, реализующей распределенное хранение ключей шифрования для API-ключей банков, на основе схемы разделения секрета Шамира [2]. В отличие от известных аналогов, предлагаемое решение адаптировано для веб-ориентированных архитектур на базе отечественного программного обеспечения и не требует приобретения специализированного аппаратного обеспечения.

Объектом исследования выступила распределенная информационная система, предназначенная для интеграции банковских сервисов в CRM-платформы (Битрикс, AmoCRM и др.). Архитектура системы включает три основных компонента (рис. 1):

1. **Пользовательский сервер** (обрабатывает запросы клиентов, управляет процессом получения данных, не хранит чувствительную информацию).

2. **Сервер работы с API** (реализует спец-

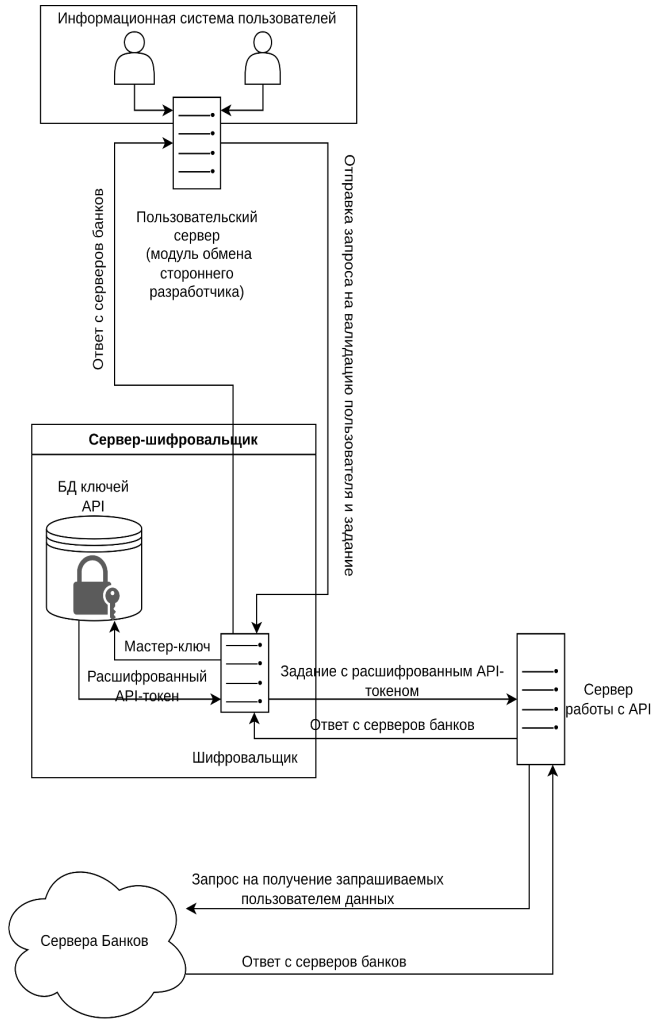

Рис. 1. Архитектура распределенной ИС, предназначенной для интеграции банковских сервисов в CRM-платформы

ифику обращений к API различных банков с целью генерации сертификатов, формирования запросов).

3. **Сервер-шифровальщик** (отвечает за хранение API-токенов в зашифрованном виде, валидацию запросов и инициирование работы сервера работы с API).

В текущей архитектуре сервер-шифровальщик локально хранит ключи для расшифровки базы данных (БД) API-ключей. Такая архитектура имеет критическую уязвимость: получение полного доступа к серверу-шифровальщику позволяет злоумышленнику извлечь ключи расшифровки БД и получить доступ к API-токенам пользователей [3]. Несмотря на валидацию запросов, защита данных ИС зависит от безопасности единственного узла.

Сравнение с существующими аналогами показывает следующие недостатки традиционных подходов:

• **аппаратные модули безопасности (HSM)** – обеспечивают высокий уровень защиты, но требуют значительных затрат и сложной интеграции с веб-приложениями;

• **централизованные серверы ключей** – упрощают управление, но создают единую точку отказа и могут стать основной целью для атакующих.

Предложенное в данной работе решение лишено указанных недостатков: оно реализуется программными средствами, не требует специализированного оборудования и обеспечивает распределенное хранение ключей без зависимости от внешних сервисов.

Для устранения выявленной уязвимости предложена модификация архитектуры, исключающая хранение полного ключа шифрования на одном узле. В основе метода лежит схема разделения секрета Шамира, использующая интерполяционные многочлены Лагранжа над конечными полями [2].

Пусть S — секрет (ключ шифрования БД),

который необходимо разделить. Выбираются параметры k (пороговое значение количества частей для восстановления), n (общее количество частей) и простое число q, определяющее конечное поле F_q, где $k \le n < q-1$.

Строится многочлен степени $k-1$:

$$f(x) = c_{k-1}x^{k-1} + c_{k-2}x^{k-2} + ... + c_2x^2 + c_1x + S(mod\ q), \quad (1)$$

где $f(0)=S$, а коэффициенты $c_1,\ldots, c_{k-1}$ выбираются случайным образом из F_q, причем $c_{k-1} \neq 0$.

Для разделения секрета генерируется n различных точек $(x_i, f(x_i))$, где $x_i \neq 0$. Каждая точка передается на хранение в отдельный узел системы (базу данных частей ключей).

Восстановление секрета возможно при наличии любых k точек. Используя интерполяцию Лагранжа, восстанавливается многочлен $f(x)$:

$$f(x) = \Sigma^k_{i=1} l_i(x)(mod\ q) \quad (2)$$

где базисные многочлены Лагранжа вычисляются как:

$$l_i(x) = f(x_i) \cdot (h_i(x_i))^{-1} \cdot h_i(x)(mod\ q) \quad (3)$$

$$h_i(x) = \Pi^k_{j=1, j\neq i}(x-x_j)(mod\ q) \quad (4)$$

Секрет восстанавливается вычислением $S = f(0)(mod\ q)$ [2].

В предложенной архитектуре сервер-шифровальщик не хранит ключи БД в явном виде. Вместо этого внедрены несколько серверов БД частей ключей. Ключ шифрования для каждой ячейки основной БД разделяется на n частей и распределяется между этими серверами (рис. 2).

Преимущества новой архитектуры по сравнению с аналогами:

1. **Устойчивость к компрометации** –

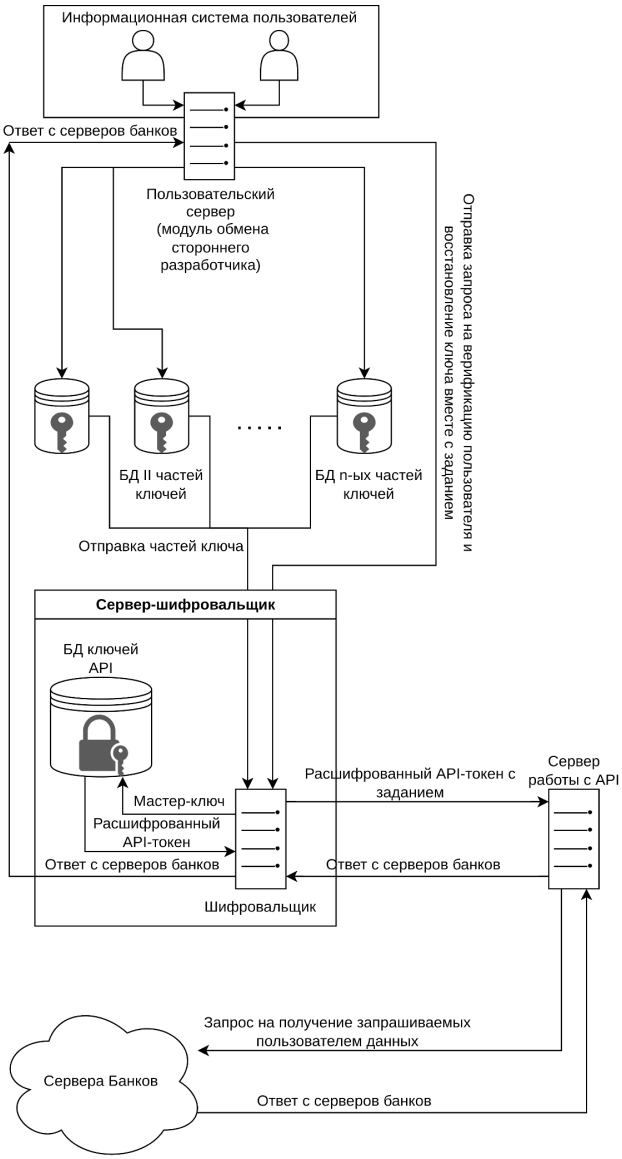

Рис. 2. Архитектура распределенной ИС с разделенной БД ключей шифрования

взлом одного сервера частей ключей не дает злоумышленнику никакой информации о секрете (при условии $k>1$). В отличие от HSM, где компрометация устройства ведет к утечке всех ключей, в предложенной схеме необходимо контролировать как минимум k узлов [4].

2. **Отказоустойчивость** – система сохраняет работоспособность при недоступности части узлов, если количество доступных частей не менее k. Это преимущество отсутствует в централизованных решениях.

3. **Экономическая эффективность** – решение реализуется на стандартном серверном оборудовании без необходимости приобретения специализированных криптографических средств.

Недостатком предложенной архитектуры распределенной ИС с разделенной БД ключей шифрования является переход сервера-шифровальщика из состояния «без сохранения» в состояние «с сохранением» данных, так как для операции расшифровки требуется процедура сбора и восстановления ключа, что предполагает необходимость дополнительных вычислений. Однако для обеспечения безопасности хранения ключей подобный компромисс является обоснованным.

Для оценки эффективности предложенной архитектуры разработан демонстрационный стенд. Выбор средств разработки обусловлен требованиями к использованию сертифицированных средств защиты информации (СЗИ) и импортозамещению:

1. **Операционная система** Astra Linux Special Edition 1.8 [5].

2. **Веб-сервер** Apache 2.4.65.

3. **Язык программирования** PHP 8.2.29 [6].

4. **СУБД** PostgreSQL 15.14.

Особое внимание уделено используемым криптографическим средствам. Для шифрования данных выбран алгоритм «Кузнечик» (ГОСТ Р 34.12-2015) [7]. Ввиду отсутствия официальной поддержки шифра «Кузнечик» и схемы Шамира в стандартных библиотеках PHP (включая расширения КриптоПРО на момент проведения исследования), алгоритмы были реализованы автором самостоятельно с соблюдением спецификаций ГОСТ [7], [8].

Для количественной оценки предложенного решения была разработана методика проведения экспериментов, включающая следующие этапы:

1. **Подготовка тестовой среды.** Все компоненты системы развернуты на одной машине со следующей конфигурацией: 8 ядер CPU (Intel Core i5-10210U, 1.60 ГГц), 16 ГБ ОЗУ, диск SSD NVMe 512 ГБ, сетевой интерфейс 1 Гбит/с. Общение узлов происходит через локальную сеть. Сетевая задержка между узлами эмулирована с помощью утилиты tc (trafficcontrol) и составляет 1 – 3 мс, что соответствует реалистичным условиям локального ЦОД.

2. **Формирование тестовой выборки**. Для каждой комбинации параметров (k, n) проведено $N=1000$ независимых измерений времени выполнения операций разделения и восстановления ключа. Ключ шифрования размером 256 бит выбирался случайным образом. Порядок поля q выбирался равным $2^{521} - 1$. Это наименьшее простое число Мерсенна, которое больше, чем 2^{256}.

3. **Измерение временных характеристик.** Замеры времени выполнялись с использованием функции hrtime() языка программирования PHP с точностью до 1 мкс [6]. Для каждой серии измерений вычислялись выборочное среднее $\overline{x}$, выборочное среднеквадратическое отклонение s и доверительный интервал для математического ожидания (при доверительной вероятности $\gamma = 0,95$).

4. **Статистическая обработка результатов.** Проверка по критерию Шапиро–Уилка с $p>0,05$ показала, что распределение времени выполнения операций асимптотически нормально, поэтому доверительный интервал для математического ожидания можно рассчитать по формуле:

$$\overline{x} \pm t_{\gamma,\nu} \cdot \frac{s}{\sqrt{N}}, \qquad (5)$$

где $t_{\gamma,\nu}$ – квантиль распределения Стьюдента; $\nu = N-1$ – количество степеней свободы.

При $\gamma = 0,95$ и $N=1000$ получим $t_{\gamma,\nu} \approx 1.96$[6].

5. **Контроль внешних факторов**. Перед началом измерений выполнялась предварительная серия из 100 итераций для исключения влияния кэширования кода и инициализации соединений. Во время тестирования нагрузка на хост-систему минимизировалась, фоновые процессы отключались.

В табл. 1 приведены результаты замеров времени разделения и восстановления ключа при различных значениях параметров k и n с указанием 95% доверительных интервалов.

Анализ данных табл. 1 показывает, что время операций растет приблизительно линейно в зависимости от параметра n, что говорит о задержках при сборе частей ключей на сервер-шифровальщике, так как базы данных опрашиваются последовательно. При

Таблица 1

Временные характеристики операций схемы Шамира (95% доверительные интервалы)

Параметры (k, n)	Время разделения ключа, мс	Время восстановления ключа, мс	Размер одной доли, бит
(2, 3)	125.67 ± 0.44	131.36 ± 1.03	521
(3, 5)	180.53 ± 0.54	171.03 ± 1.53	521
(4, 6)	201.17 ± 0.82	196.08 ± 1.96	521
(5, 7)	234.24 ± 0.69	220.94 ± 2.17	521

больших значениях $k \geq 30$ следует ожидать квадратичный рост времени восстановления ключа ввиду алгоритмической сложности интерполяции Лагранжа. Но такое количество серверов не имеет практического смысла. Узкие доверительные интервалы (относительная погрешность не превышает 1,5%) свидетельствуют о стабильности работы алгоритма и достаточном объеме выборки для статистических выводов [9].

Дополнительно были измерены накладные расходы на шифрование/расшифрование данных алгоритмом «Кузнечик» в программной реализации на PHP (100 тестов, входные данные — строка 16 кБ):

• Скорость шифрования: 17.30 ± 0.12кБ/с (95% ДИ);

• Скорость расшифрования: 16.80 ± 0.07 кБ/с (95% ДИ).

Применение новой архитектуры ИС привело к увеличению времени отклика на 100 – 200 мс по сравнению с архитектурой без схемы Шамира. Это связано, в основном, с сетевыми задержками при опросе серверов частей ключей.

Предложенное решение обеспечивает сопоставимый уровень защищенности при значительно меньших затратах времени на интеграцию по сравнению с известными решениями. Например, время доступа к ключу в HSM составляет 10 – 50 мс, а в разработанной архитектуре – 150 – 200 мс. Кроме того, стоимость внедрения предложенного решения гораздо ниже благодаря использованию имеющегося серверного оборудования.

Предложенная архитектура обеспечивает существенное повышение защищенности, поскольку для компрометации ключа шифрования злоумышленнику необходимо получить контроль как минимум над k серверами хранения долей, что экспоненциально снижает вероятность успешной атаки [4], но предполагает увеличение объемов вычислений.

Повышение защищенности ключей шифрования обеспечивается тем, что для компрометации ключа злоумышленнику необходимо получить контроль как минимум над k серверами хранения долей, что экспоненциально снижает вероятность успешной атаки [4]. Однако предложенная архитектура предполагает увеличение объемов вычислений.

Все компоненты системы реализованы на базе отечественного ПО ОС Astra Linux, СУБД PostgreSQL, что обеспечивает соответствие требованиям регуляторов в области импортозамещения и использования сертифицированных СЗИ [5]. Предложенное решение может быть интегрировано в действующие CRM-платформы (Битрикс, AmoCRM) без необходимости полной замены ПО, что минимизирует затраты на внедрение.

Ограничением подхода является увеличение объема хранимых данных: вместо одного ключа система хранит n долей, каждая из которых по размеру равна исходному секрету. Однако для ключей шифрования (типичный размер 256 бит) данное увеличение несущественно по сравнению с общим объемом данных системы.

Заключение

Таким образом, в ходе исследования разработана архитектура ИС, обеспечивающая повышенную защищенность API-ключей банковских сервисов. Внедрение схемы разделения секрета Шамира позволяет устранить риск массовой утечки данных при компрометации отдельного сервера инфраструктуры [2]. Реализация ИС на базе отечественного ПО с применением алгоритма «Кузнечик» обеспечивает соответствие требованиям ГОСТов [5], [7], [8].

В отличие от рассмотренных аналогов (HSM, централизованные серверы ключей), предложенная архитектура сочетает высокий уровень защищенности с возможностью интеграции в веб-архитектуры без дополнительного оборудования. Статистический ана-

лиз предложенного метода разделений ключей показал практическую применимость предложенного решения, так как накладные расходы на операции разделения и восстановления ключей не превышают 235 мс (с доверительной вероятностью 0,95), что допустимо для задач интеграции с внешними банковскими сервисами [9].

В перспективе предложенную архитектуру распределенной ИС с разделенной БД ключей шифрования можно адаптировать для динамического изменения параметров *k* и *n* без перевыпуска всех долей секрета и интегрировать с аппаратными доверенными платформами.

Литература

1. Соколов С.С., Новоселов Р.Ю., Митрофанова А.В. Методы обеспечения доступности информации в высоконагруженных информационных системах // Вестник УрФО № 2(28) / 2018, с. 31–35.

2. Шамир А. Как разделить секрет, чтобы обходиться без защищенного канала связи // Проблемы передачи информации. 1981. Т. 17. № 3. С. 3–14.

3. Повышев А.А., Соколов А.Н., Мищенко Е.Ю. Универсальная классификация угроз безопасности информации и её применение для разработки модели угроз и оценки рисков // Вестник УрФО № 3(49) / 2023, с. 68–80.

4. Шевяков И.А., Соколов А.Н. Метод определения уровня защищённости критических узлов информационной сети транспортного средства // Вестник УрФО № 3(45) / 2022, с. 83–91.

5. Операционная система специального назначения «Astra Linux Special Edition». Руководство по комплексу средств защиты. URL: https://wiki.astralinux.ru/download/attachments/371802537/1.8.3_Ruk_KSZ_1.pdf (дата обращения: 10.03.2026).

6. PHP: Hypertext Preprocessor. Official Documentation. URL: https://www.php.net/manual/ru/ (дата обращения: 01.03.2026).

7. ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры. М.: Стандартинформ, 2015.

8. ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хеширования. М.: Стандартинформ, 2012.

9. ГОСТ Р ИСО/МЭК 17025-2019. Общие требования к компетентности испытательных и калибровочных лабораторий. М.: Стандартинформ, 2019.

References

1. Sokolov S.S., Novoselov R.Ju., Mitrofanova A.V. Metody obespechenija dostupnosti informacii v vysokonagruzhennyh informacionnyh sistemah // Vestnik UrFO № 2(28) / 2018, s. 31–35.

2. Shamir A. Kak razdelit' sekret, chtoby obhodit'sja bez zashhishhennogo kanala svjazi // Problemy peredachi informacii. 1981. T. 17. № 3. S. 3–14.

3. Povyshev A.A., Sokolov A.N., Mishhenko E.Ju. Universal'naja klassifikacija ugroz bezopasnosti informacii i ejo primenenie dlja razrabotki modeli ugroz i ocenki riskov // Vestnik UrFO No 3(49) / 2023, s. 68–80.

4. Shevjakov I.A., Sokolov A.N. Metod opredelenija urovnja zashhishhjonnosti kriticheskih uzlov informacionnoj seti transportnogo sredstva // Vestnik UrFO No 3(45) / 2022, s. 83–91.

5. Operacionnaja sistema special'nogo naznachenija «Astra Linux Special Edition». Rukovodstvo po kompleksu sredstv zashhity. URL: https://wiki.astralinux.ru/download/attachments/371802537/1.8.3_Ruk_KSZ_1.pdf (data obrashhenija: 10.03.2026).

6. PHP: Hypertext Preprocessor. Official Documentation. URL: https://www.php.net/manual/ru/ (data obrashhenija: 01.03.2026).

7. GOST R 34.12-2015. Informacionnaja tehnologija. Kriptograficheskaja zashhita informacii. Blochnyeshifry. M.: Standartinform, 2015.

8. GOST R 34.11-2012. Informacionnaja tehnologija. Kriptograficheskaja zashhita informacii. Funkcija heshirovanija. M.: Standartinform, 2012.

9. GOST R ISO/MJeK 17025-2019. Obshhie trebovanija k kompetentnosti ispytatel'nyh i kalibrovochnyh laboratorij. M.: Standartinform, 2019.

ЗУЕВ Александр Денисович, студент кафедры защиты информации ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)». 454080, г. Челябинск, проспект Ленина, 76. E-mail: zuev.ad@icloud.com

ЗЮЛЯРКИНА Наталья Дмитриевна, доктор физико-математических наук, доцент, профессор кафедры защиты информации ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)». 454080, г. Челябинск, проспект Ленина, 76. E-mail: ziuliarkinand@susu.ru

СОКОЛОВ Александр Николаевич, кандидат технических наук, доцент, заведующий кафедрой защиты информации ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)». 454080, г. Челябинск, проспект Ленина, 76. E-mail: sokolovan@susu.ru

ZUEV Aleksandr Denisovich, Student of the Department of Information Security, Federal State Autonomous Educational Institution of Higher Education "South Ural State University (national research university)". 454080, Chelyabinsk, Lenin prospekt, 76. E-mail: zuev.ad@icloud.com

ZYULYARKINA Natalia Dmitrievna, Doctor of Physical and Mathematical Sciences, Associate Professor of the Department of Information Security, Federal State Autonomous Educational Institution of Higher Education "South Ural State University (national research university)". 454080, Chelyabinsk, Lenin prospekt, 76. E-mail: ziuliarkinand@susu.ru

SOKOLOV Aleksandr Nikolaevich, Candidate of Technical Sciences, Associate Professor, Head of the Department of Information Security, Federal State Autonomous Educational Institution of Higher Education "South Ural State University (national research university)". 454080, Chelyabinsk, Lenin prospekt, 76. E-mail: sokolovan@susu.ru