СХЕМА КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ДАННЫХ ОБЛАЧНЫХ ХРАНИЛИЩ С ВОЗМОЖНОСТЬЮ РАСШИРЕНИЯ КВАНТОВОЗАВИСИМЫМИ КЛЮЧАМИ И ПОСТКВАНТОВЫМИ АЛГОРИТМАМИ

Минаков С. С.[1], Тихов С. В.[2], Цупак А. А.[3]

DOI: 10.21681/2311-3456-2025-6-88-100

Цель исследования: *разработка схемы криптографической защиты информации в облачных хранилищах с использованием стандартизированных и перспективных режимов блочных шифров, постквантовых алгоритмов шифрования и квантовозависимых ключей.*

Метод(ы) исследования: *системный анализ угроз безопасности информации при её обработке и хранении с использованием прикладных протоколов доступа к облачным хранилищам. Синтез криптографических механизмов и форматов для построения автоматизированной гибридной криптографической схемы обработки информации на стороне потребителя облачных услуг с использованием симметричных и асимметричных шифров.*

Результат(ы) исследования: *представлено развитие криптографической схемы «Утро-1» для обеспечения защиты информации в прикладных протоколах доступа облачных хранилищ. Описаны форматы, функции и логика шифрования, проведены практические испытания программной реализации. Даны пояснения по условиям использования в криптографической схеме постквантовых алгоритмов и/или квантовозависимых ключей.*

Научная новизна: *получены новые научно-технические результаты в области защиты данных в облачных хранилищах с использованием российских и перспективных иностранных криптографических средств и методов; практическая ценность состоит в развитии криптографической схемы защиты информации от НСД с использованием гибридной композиции симметричных и асимметричных шифров.*

Ключевые слова: *шифрование, облачный сервис, сетевой доступ, прикладной протокол, безопасность информации, выработка ключей.*[4567]

Введение

Современный уровень и задачи развития систем информатизации, цифровая трансформация государственного управления привели к переводу на автоматизированную и интеллектуальную обработку большинства процессов управления и обработки информации в госуправлении, негосударственных организациях, на предприятиях промышленности и транспорта. Из таких систем сложилась и нормативно закреплена сфера критической информационной инфраструктуры (КИИ).

При этом значительный пласт автоматизированных и информационных систем, в том числе и информационные системы персональных данных, теперь разворачиваются в сторонних центрах обработки и хранения данных (далее – ЦОД). Более того, в рамках реализации нескольких государственных программ Правительством Российской Федерации одобрены концепции перевода государственных информационных ресурсов в ЦОДы[4] и развёртывания государственной единой облачной платформы (платформа ГЕОП «Гособлако»), с 2019 г. начаты соответствующие эксперименты[5] и к 2021 г. создана единая цифровая платформа Российской Федерации «ГосТех» (платформа ГосТех[6,7]).

При использовании сторонних для защищаемой системы сервисов облачных вычислений существенно изменяется модель угроз и нарушителя, во многом утрачивается подконтрольность обрабатываемой и хранимой информации, что зачастую не позволяет наделять сторону облачных сервисов требуемыми гарантиями доверия и считать достаточными рубежи защиты только на базе криптографических протоколов взаимодействия с облачным хранилищем (например, протоколы TLS, IPSec).

Таким образом, необходимо при построении сегментов системы с различным уровнем доверия

1 Минаков Сергей Сергеевич, старший научный сотрудник, ФГКНУ «Академия криптографии Российской Федерации». г. Москва, Россия. E-mail: ss_minakov@mail.ru
2 Тихов Станислав Вячеславович, ведущий специалист ООО НТП «Криптософт». г. Пенза, Россия. E-mail: tik.stanislav2015@yandex.ru
3 Цупак Алексей Александрович, доктор физико-математических наук, доцент, доцент кафедры «Математика и суперкомпьютерное моделирование» ФГБОУ «Пензенский государственный университет». г. Пенза, Россия. E-mail: altsupak@yandex.ru

4 Распоряжение Правительства Российской Федерации от 07.10.2015 № 1995-р (ред. от 18.10.2018).
5 Постановление Правительства РФ от 28.08.2019 № 1114.
6 Распоряжение Правительства РФ от 21.10.2022 № 3102-р.
7 «Концепция обеспечения информационной безопасности единой цифровой платформы Российской Федерации «ГосТех» (утв. приказом Минцифры России от 12.01.2023, № 7).

и при наличии сторонних облачных сервисов переходить от криптографического протокола к криптографической схеме [1], в которой сторона, реализующая техническую службу облачного хранилища, не может гарантировать конфиденциальность и целостность данных пользователя, обеспечивать подтверждение корректности реализаций механизмов обеспечения защиты информации при реализации облачных вычислений.

Рассмотрим вариант криптографической защиты с использованием прикладных протоколов доступа и развитие криптографической схемы «Утро-1» [1], уточним криптографические механизмы защиты данных, передаваемых в облачные сервисы категории Data Storage as a Service (далее — STaaS / DsaaS), для повышения эффективности защиты информации при использовании облачных вычислений и стандартизированных интерфейсов и протоколов управления облачными данными CDMI (англ. – Cloud Data Management Interface), включая прикладные протоколы *AWS Simple Storage Service* (Amazon Web Services S3 protocol), *WebDAV* и *Calendaring Extensions to WebDAV (CalDAV protocol)* для доступа к ресурсам облачных хранилищ [2].

Внедрение новых суперкомпьютеров и квантовых вычислителей [3–5] позволяют говорить о конкретной технической реализации эффективных квантовых алгоритмов на квантовых компьютерах или массовой эмуляции их при распределенных вычислениях [6]. В данной статье авторы рассматривают два подхода к противодействию «квантовой угрозе» (атаки на асимметричные и симметричные блочные алгоритмы шифрования) в рамках задачи о защищенном облачном хранении данных: использование квантовозависимой ключевой информации, полученной с использованием систем квантового распределения ключей и использование методов «постквантовой» криптографии наряду с классическими стандартизованными блочными шифрами.

Структуры данных в прикладных протоколах доступа к облачным хранилищам файлов

Рассмотрим структуры данных в трех типовых прикладных протоколах доступа к облачному хранилищу WebDAV, S3 и CalDAV. Отметим, что ни WebDAV, ни S3, ни CalDAV не являются полноценными сетевыми протоколами, а скорее представляют собой наборы расширений и дополнений к протоколу HTTP. Тем не менее, опуская тонкости терминологии, будем использовать слово «протокол» (наряду с «технологией» и «интерфейсом») применительно к этим трем технологиям доступа к облачному хранилищу.

Являясь расширением протокола прикладного уровня HTTP, протоколы доступа WebDAV, AWS S3, CalDAV имеют много общего в части базовой технологии передачи и получения данных. В то же время, они различаются между собой в части используемых методов и структур данных в HTTP-запросах.

В облачных сервисах хранения данных ***технология WebDAV*** используется в качестве сетевой файловой системы, а также в качестве протокола для удаленного доступа к ее объектам (файл-контейнерам) и управления ими. К основным возможностям WebDAV относят: механизм блокировок на доступ к объектам хранилища, средства для создания, редактирования и удаления свойств (метаданных) объектов, а также использование именованных областей, представляющих коллекцию объектов и выполняющих функцию, аналогичную каталогам в файловой системе.

Согласно спецификации RFC 4918 для удаленного доступа и управления объектами хранилища протокол WebDAV предоставляет «унаследованные» от HTTP команды PUT, GET и DELETE, а также семь новых команд: PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK и UNLOCK. Команды PUT и GET применяются, соответственно, для записи и чтения данных в/из файл-контейнер(а). Остальные команды предназначены для управления уже размещенными в хранилище объектами (переименование или перемещение, разграничение прав доступа и др.). Таким образом, потенциально уязвимые (содержащие конфиденциальные сведения) данные пользователя передаются только с командами PUT и GET.

Запрос на запись, например, текстовой строки «Hello world» (в кодировке UTF-8) в файл-контейнер *Obj* облачного хранилища имеет вид

```
PUT /Obj HTTP/1.1
Host: www.example.com
Content-Type: text/plain
Content-Length: 12
Hello world
```

Первые четыре строки – это типовые заголовки HTTP-запроса, а последняя строка представляет собой тело запроса с данными для записи в файл-контейнер *Obj*. При успешном выполнении запроса сервер возвращает ответ вида

```
HTTP/1.1 201 Created
Content-Length: 12
Date: Sat, 1 May 2025 09:32:12 GMT
```

Запрос на чтение данных из файл-контейнера *Obj* имеет вид

```
GET /Obj HTTP/1.1
Host: www.example.com
Accept: */*
```

При успешном выполнении запроса сервер отвечает:

```
HTTP/1.1 200 OK
Content-Type: text/plain;
charset=utf-8
Content-Length: 12
Hello world
```

DOI: 10.21681/2311-3456-2025-6-88-100

Для успешного выполнения представленных выше запросов, последние должны включать некоторые дополнительные заголовки. К их числу относится заголовок *Authorization*, содержащий информацию для аутентификации стороны, отправляющей запрос облачному сервису.

В представленных примерах тела запросов содержат текстовые данные. В общем же случае они могут содержать произвольные бинарные данные, в частности, любые датаграммы, инкапсулирующие данные пользователя. Это позволяет реализовать «защищенную версию» WebDAV (с размещением данных в файл-контейнерах в зашифрованном виде) без внесения каких-либо изменений или дополнений в оригинальный протокол доступа к облачному хранилищу.

Протокол S3 предназначен для доступа к облачным хранилищам, построенным по одноименной технологии, – так называемым, *S3-хранилищам*. Последние представляют собой плоские (не иерархические) хранилища, где все объекты являются равнозначными и другой, встроенной системы разграничения доступа между ними нет.

Каждый объект в S3-хранилище состоит из трех компонент: уникального идентификатора (строка, служащая для однозначной идентификации объекта и прямого доступа к нему по URL-адресу), метаданных (дополнительные атрибуты объекта, такие, как размер, тип файла и др.) и содержимого. Для удобной работы объекты хранилища можно объединить в так называемые *бакеты* (англ. buckets). Бакет получает уникальный идентификатор, входящий в URL каждого объекта, включенного в бакет.

Для управления объектами хранилища протокол S3 предоставляет обширный интерфейс, включающий несколько сотен различных операций (HTTP-запросов). Как и в случае с протоколом WebDAV, конфиденциальные данные пользователя передаются в S3-хранилище лишь в запросах на запись (*PutObject*) и чтение (*GetObject*) данных в/из объекта хранилища, тогда как остальные запросы оперируют либо со служебными, либо с уже размещенными в хранилище данными.

Авторы не приводят здесь форматную структуру запросов *PutObject* и *GetObject*, поскольку они полностью совпадают с форматами запросов *PUT* и *GET* протокола WebDAV, соответственно, за исключением набора HTTP-заголовков, необходимых для успешного выполнения запроса.

Протокол CalDAV определяет стандартный способ доступа, управления и обмена информацией о календарях и расписаниях в формате *iCalendar*. В отличие от протоколов WebDAV и S3, позволяющих размещать в хранилище произвольные бинарные данные пользователя, протокол CalDAV предназначен для работы с информацией, имеющей специальный формат, определенный в RFC 5545.

Данные о календарях и расписаниях представляются *объектом iCalendar* вида:

```
BEGIN:VCALENDAR
icalbody
END:VCALENDAR
```

Первая и последняя строки являются константными и указывают на начало и конец объекта iCalendar. Тело объекта *icalbody* состоит из последовательности *свойств календаря* и одного или нескольких компонент календаря. Свойства календаря – это атрибуты, которые применяются к объекту календаря в целом. *Компоненты календаря* – это наборы свойств, которые выражают определенную семантику календаря. Так, компоненты календаря могут определять событие, задачу, запись в журнале и информацию о свободном/занятом времени.

Объект iCalendar организован в виде отдельных строк, называемых в соответствии с RFC 5545 *контентными строками* (англ. content lines), длиной не более 75 байт, разделенных последовательностью символов CRLF. Каждая такая строка соответствует тому или иному свойству календаря или компоненты календаря и имеет вид «name:value», где name – название свойства, а value – его значение. Ниже представлен пример объекта iCalendar из документа RFC 5545:

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//CalDav client//EN
BEGIN:VEVENT
UID:123
DTSTART:20250101T000000Z
DTEND:20250101T010000Z
SUMMARY: New Year Party
END:VEVENT
END:VCALENDAR
```

Вторая и третья строки – это свойства календаря: VERSION указывает на версию спецификации формата, PRODID указывает идентификатор приложения, генерирующего данные календаря. Строки с четвертой по девятую соответствуют компоненте календаря VEVENT, представляющей событие с четырьмя свойствами: UID – его уникальный идентификатор, DTSTART и DTEND указывают дату и время начала и окончания события, соответственно, SUMMARY содержит описание события.

Свойства объектов iCalendar строго типизированы. Спецификация RFC 5545 определяет четырнадцать типов значений, в том числе бинарные, текстовые и числовые типы данных, адреса пользователей календаря, несколько типов данных для хранения даты и времени и др. Так, для идентификации значений,

содержащих точную календарную дату и время суток, используется, как правило, тип DATE-TIME. Формат значений DATE-TIME основан на представлении дат и времени, описанном в стандарте ГОСТ Р 7.0.64-2018 (ИСО 8601:2004). В примере выше свойства DSTART и DTEND хранят данные типа DATE-TIME.

Строгая типизированность свойств объектов iCalendar делает невозможным применение к некоторым из них криптографических преобразований; это справедливо, например, для свойств, содержащих дату и время, периодичность наступления события (тип данных *RECUR*). Предлагаемые в данной работе механизмы криптографической защиты данных календаря учитывают эти ограничения.

Рассмотрим протокол CalDAV. Как следует из его названия, CalDAV является производным от WebDAV и для хранения объектов iCalendar использует иерархическую структуру DAV-хранилища. Согласно спецификации RFC 4791, календарь CalDAV представляет собой DAV-коллекцию (каталог) с определенной структурой; каждая такая коллекция-календарь включает набор так называемых *ресурсов объекта календаря* (англ. *calendar object resources*) – файл-контейнеров, которые содержат свойства (набор контентных строк) либо всего календаря, либо одной из его компонент. При этом CalDAV, как протокол доступа, инкапсулирует организацию данных в хранилище и предоставляет инструменты для доступа и управления данными календаря на уровне объектов iCalendar, а не отдельных файлов-ресурсов объектов iCalendar. Так, например, запрос на добавление в календарь нового события имеет вид

```
PUT /calendars/events/
    /qwue23489.ics HTTP/1.1
Host:www.example.com
Content-Type: text/calendar
Content-Length: 234
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//CalDAV client//EN
BEGIN:VEVENT
UID:123
DTSTAMP:20241111T000000Z
DTSTART:20250101T000000Z
DTEND:20250101T010000Z
SUMMARY: New Year Party
END:VEVENT
END:VCALENDAR
```

Как видно из этого примера, тело запроса представляет объект iCalendar. В случае успешного выполнения запроса служба облачного сервиса CalDAV создает в каталоге-календаре файл-контейнер и записывает в него свойства новой компоненты (события) календаря.

Построение гибридной криптографической схемы с использованием сертификатов безопасности

Предлагаемая авторами криптографическая схема (КС) защиты данных, размещаемых в облачных хранилищах, является гибридной в том смысле, что она включает как композиции симметричных алгоритмов шифрования для выполнения преобразований и имитозащиты данных, так и механизмы асимметричной криптографии для защищенного обмена ключевым материалом [2]. В части вариантов реализации симметричной криптографии на основе блочных шифров предлагаемая криптосхема во многом наследует КС «Утро» (вариант «Утро-1»), подробно изложенной в работе [1].

Композиция алгоритмов асимметричных криптографических механизмов в схеме авторов работы строится вокруг протокола Диффи-Хеллмана, но реализованного на эллиптических кривых.

Для изложения предлагаемой криптографической схемы удобно рассмотреть типовой сценарий использования облачного хранилища: сторона А обменивается данными со стороной Б, для чего создает в облачном хранилище новый файл-контейнер Obj и размещает в нем данные в защищенном виде.

Предполагается, что сторона А на момент размещения данных в облачном хранилище имеет валидный (верифицированный доверенной третьей стороной) сертификат безопасности стороны Б, содержащий открытый ключ $y \cdot P$ последней.

Размещение данных в файл-контейнере облачного хранилища

Шаг 1. Для каждого нового файл-контейнера облачного хранилища данных и, в частности, файл-контейнера для Obj, генерируется пара 256-битных ключей: мастер-ключ K_E для выполнения криптографических преобразований данных и мастер-ключ K_M для выполнения имитозащиты данных.

Шаг 2. При помощи секретного ключа x стороны А и открытого ключа $y \cdot P$ стороны Б формируется ключ шифрования KEK_{VKO} в соответствии с выбранным алгоритмом согласования ключей VKO.

Шаг 3. Используя полученный на шаге 2 ключ шифрования KEK_{VKO}, осуществляется процедура экспорта материала ключей K_E и K_M согласно выбранному алгоритму экспорта ключей. В результате получается зашифрованный материал ключей K_E и K_M, а также значения имитозащитных вставок $MAC(K_E)$, $MAC(K_M)$, которые затем используются в процедуре импорта для проверки корректности восстановления (расшифрования) ключевого материала.

Шаг 4. Размещаемый на облаке файл-контейнер Obj представляется в виде набора из N логических последовательно нумерованных равных (возможно, за исключением последнего) фрагментов данных. Пусть Obj_k обозначает k-ый фрагмент файл-контейнера

Obj. При этом размер отдельного фрагмента данных определяется требованиями, предъявляемыми к максимальному объему информации, который может быть обработан на одном ключе. На основе мастер-ключей K_E и K_M формируются два множества (базиса) производных ключей $\{K_E^k\}_{k=1}^N$ и $\{K_M^k\}_{k=1}^N$, соответственно. Ключ K_E^k предназначен для выполнения криптографических преобразований фрагмента Obj_k. Ключ K_M^k предназначен для вычисления значения имитозащитной вставки фрагмента Obj_k.

Шаг 5. Сторона А зашифровывает каждый фрагмент Obj_k, $k = (1,N)$ – выбранным алгоритмом блочного шифрования на соответствующем ключе из базиса $\{K_E^k\}_{k=1}^N$ и вычисляет значение имитозащитной вставки $MAC(Obj_k)$ на соответствующем ключе из базиса $\{K_M^k\}_{k=1}^N$.

Шаг 6. Сторона А последовательно загружает в облачное хранилище зашифрованные фрагменты Obj_k, а также значения $MAC(Obj_k)$. Формат файл-контейнера с защищенными данными обсуждается ниже.

Шаг 7. Полученные на шаге 3 экспортированные (зашифрованные) материалы мастер-ключей K_E и K_M и значения $MAC(K_E)$, $MAC(K_M)$ также загружаются стороной А в облачное хранилище данных. Способы хранения указанных данных в облачном хранилище обсуждаются ниже.

Получение данных из файл-контейнера облачного хранилища

Шаг 1. Сторона Б загружает из облачного хранилища экспортированные (зашифрованные) стороной А ключи K_E и K_M и значения $MAC(K_E)$, $MAC(K_M)$.

Шаг 2. При помощи секретного ключа у стороны Б и открытого ключа $x \cdot P$ стороны А формируется ключ шифрования KEK_{VKO}, идентичный тому, что был получен на шаге 2 схемы размещения данных в файл-контейнере Obj.

Шаг 3. Используя KEK_{VKO}, осуществляется процедура импорта ключей K_E и K_M согласно выбранному алгоритму экспорта/импорта ключей. В частности, выполняется проверка совпадения вычисленных на ключе KEK_{VKO} (либо производном от KEK_{VKO} ключе) имитовставок восстановленных материалов ключей K_E и K_M и имитовставок $MAC(K_E)$, $MAC(K_M)$, загруженных на шаге 1 из облака.

Шаг 4. Сторона Б последовательно загружает из облачного хранилища зашифрованные фрагменты Obj_k файл-контейнера Obj, а также значения имитозащитных вставок $MAC(Obj_k)$.

Шаг 5. Расшифрование фрагмента Obj_k производится в обратном порядке только после успешной проверки совпадения вычисленной на соответствующем ключе имитовставки такого фрагмента Obj_k и имитовставки, полученной из облачного хранилища на шаге 4.

Используемые в схеме криптографические функции

Для выработки материала K_E и K_M в качестве источников псевдослучайных последовательностей R можно использовать псевдослучайную функцию PRF_TLS_GOSTR3411_2012_256 с длиной выхода 256 бит, определенную в рекомендациях по стандартизации Р 50.1.113-2016, или алгоритм выработки псевдослучайной последовательности R длины 256 бит, определенный в рекомендациях по стандартизации Р 1323565.1.006-2017. При этом энтропийные данные, передаваемые в эти функции в качестве параметров, предлагается получать при помощи аппаратного ДСЧ, биологического ДСЧ, или квантового ДСЧ.

В качестве секретного и открытого ключей, применяемых в изложенной выше схеме в части защищенного обмена материалами ключей K_E и K_M между несколькими пользователями, могут использоваться ключи подписи и проверки подписи согласно алгоритму ГОСТ Р 34.10-2018 с параметрами эллиптических кривых, определенными в Р 50.1.114-2016.

Для получения одинакового для сторон А и Б ключа шифрования KEK_{VKO} используется алгоритм согласования ключей VKO_GOSTR3410_2012_256, описанный в Р 50.1.113-2016, при этом параметр UKM алгоритма должен иметь длину не менее 64 бит ввиду того, что секретный и открытый ключи участников обмена являются длительный период статическими.

Для криптографических преобразований и имитозащиты материалов ключей K_E и K_M могут использоваться алгоритм экспорта/импорта ключей, описанный в Р 50.1.113-2016, или алгоритмы экспорта KExp15 и импорта KImp15, определенные в Р 1323565.1.017–2018.

Код аутентификации сообщения используется для решения задач имитозащиты передаваемых в облачное хранилище данных и должен формироваться одним из режимов алгоритма блочного шифрования $E_k^{mode}(T)$, например, ключевой функцией хеширования OMAC-ACPKM, определенной в ГОСТ 34.13-2018, либо функцией хеширования HMAC, определенной в Р 50.1.113-2016. Значения $MAC_k(Obj_k)$ разных фрагментов файл-контейнера должны вычисляться на разных ключах K_M^k, производных от мастер-ключа K_M, причем материалы ключей K_M и K_E должны быть различными. Таким образом, имитозащита данных требует увеличения вдвое размерности базиса ключей для раздельного использования ключей шифрования и ключей имитозащиты фрагментов данных в связи с существенно различной стойкостью соответствующих криптографических алгоритмов. Альтернативным решением является использование AEAD-режима работы алгоритма блочного шифрования, который обеспечивает так называемое

аутентифицируемое шифрование: шифрование и имитозащиту блока данных на одном ключе. К таким режимам относятся, например, GCM, MGM, MGM2 [7, 8].

Криптографические преобразования данных, размещаемых в файл-контейнере облачного хранилища, рекомендуется реализовать в соответствии с алгоритмами блочного шифрования «Магма» и «Кузнечик», определенными в ГОСТ Р 34.12-2018, в одном из следующих режимов работы: в режиме гаммирования с преобразованием ключа CTR-ACPKM, определенном в ГОСТ 34.13-2018, в режиме DEC (Disk Encryption with Counter Mode), описанном в рекомендациях по стандартизации Р 1323565.1.042–2022 (некоторые примечательные свойства данного режима работы блочного шифра рассмотрены в работе [9]), или новом (находящемся в процедуре стандартизации) российском режиме блочных шифров XEH, перспективным для защиты данных на системных разделах и носителях и в облачных хранилищах [10]. В качестве значения вектора инициализации iv при выполнении процедур блочного шифра в соответствующих режимах применительно к k-ому фрагменту файл-контейнера можно использовать строку (ее байтовое представление), полученную в результате конкатенации $k||k$.

Важным аспектом представленной нами криптографической схемы является вычисление производных ключей для выполнения криптографических преобразований и имитозащиты отдельных фрагментов файл-контейнеров. Функция выработки производного ключа — KDF (англ. Key derivation function) должна создавать криптографически стойкие ключи для алгоритма симметричного шифрования на основе источника первоначального ключевого материала.

Если при реализации схемы шифрование и/или имитозащита данных выполняются в режимах CTR-ACPKM и OMAC-ACPKM, то преобразование ключа (выработка производных ключей из мастер-ключа) является частью указанных режимов и выполняется при помощи функций ACPKM и ACPKM-Master с заданной частотой смены мастер-ключа, определенной в ГОСТ 34.13-2018.

В других вариантах реализации предлагаемой схемы (с использованием режимов блочного шифра XEH, MGM2, HMAC) в качестве функции вычисления производного ключа $KDF_{256}(T): V \longrightarrow V_{256}$ можно использовать как саму функцию хеширования $H_{256}(T)$, определенную в ГОСТ Р 34.11-2012, так и её производные, например, алгоритм диверсификации KDF_TREE_GOSTR3411_2012_256, определенный в Р 50.1.113-2016. Приемлемый с точки зрения производительности и криптографической стойкости от внешнего нарушителя алгоритм формирования производных ключей представлен в работе [1].

Зашифрованный материал ключа и параметры шифрования

Зашифрованный материал криптографического ключа, а также значение его имитозащитной вставки, традиционно хранят в шифрблоках-записях (т.н. *ключевых блобах*). Такие структуры данных помимо указанных данных, как правило, содержат описание опциональных параметров асимметричной схемы, например, идентификатор набора параметров эллиптической кривой (модуль эллиптической кривой, порядки группы и ее циклической подгруппы точек эллиптической кривой и т.д.), идентификаторы алгоритмов согласования и экспорта/импорта ключей.

Кроме того, для построения вариативной криптографической схемы, использующей разные СКЗИ, ключевой блоб должен быть дополнен полями, содержащими идентификаторы СКЗИ и ключевого контейнера с секретным ключом пользователя. Это необходимо, если в системе установлены несколько СКЗИ, и имеется несколько контейнеров с секретными ключами пользователя [2].

Подчеркнем, что с каждым зашифрованным файл-контейнером облачного хранилища может быть связан целый набор ключевых блобов. Так, если в конкретной реализации рассматриваемой криптографической схемы не используются AEAD-режимы работы блочного шифра, то для выработки производных ключей шифрования и имитозащиты фрагментов данных требуются два разных мастер-ключа. В этом случае необходимо экспортировать материалы обоих этих ключей. К тому же, если доступ к зашифрованным данным должен быть предоставлен группе пользователей, то материалы мастер-ключей должны быть экспортированы на нескольких ключах, полученных при помощи алгоритма согласования с использованием открытых ключей пользователей, имеющих право доступа к зашифрованным данным, и секретного ключа владельца данных (того, кто размещает данные в файл-контейнере облачного хранилища).

Набор ключевых блобов, связанных с некоторым файл-контейнером хранилища, удобно представить в виде структуры данных типа словарь «ключ-значение». Например, пусть в файл-контейнере Obj размещаются данные в защищенном виде; шифрование и имитозащита выполняются алгоритмом «Магма» в режимах CTR-ACPKM и OMAC-ACPKM, соответственно; при этом доступ к данным должен быть предоставлен еще двум пользователям. В табл. 1 представлен словарь с набором блобов, отвечающий описанной ситуации.

Таблица 1.

Словарь, хранящий набор связанных с файл-контейнером облачного хранилища ключевых блобов

Ключ	Значение
cek-alg	GR3412M/CTR-ACPKM
mac-alg	GR3412M/ACPKM-OMAC
key-wrap-alg	50.1.113-2016
kbl-count	6
enc-kbl-owner	...
mac-kbl-owner	...
enc-kbl-user-1	...
mac-kbl-user-1	...
enc-kbl-user-2	...
mac-kbl-user-2	...

Ключи cek-alg, mac-alg, key-wrap-alg хранят идентификаторы используемых в схеме алгоритмов (и режимов) блочного шифра, согласования и экспорта/импорта ключей. Ключ kbl-count хранит количество ключевых блобов, связанных с файл-контейнером. Ключи enc-kbl-owner, mac-kbl-owner, enc-kbl-user-1, mac-kbl-user-1, enc-kbl-user-2, mac-kbl-user-2 хранят непосредственно ключевые блобы, представленные в виде base64-строк.

Подобный словарь ключевых блобов файл-контейнера Obj можно разместить в облачном хранилище либо в метаданных файл-контейнера Obj, либо в отдельном файл-контейнере Obj_S с идентификатором, полученным при помощи конкатенации идентификатора Obj и некоторого зарезервированного константного суффикса.

Формат файл-контейнера облачного хранилища с защищенными данными

При размещении данных в некотором файл-контейнере Obj облачного хранилища, они загружаются туда пофрагментно. При этом к каждому фрагменту Obj_k файл-контейнера Obj применяются криптографические преобразования и для каждого фрагмента рассчитывается либо имитовставка $MAC(Obj_k)$, либо тег аутентификации $TAG(Obj_k)$ (при использовании аутентифицированного шифрования). В результате для каждого фрагмента Obj_k, $k = \overline{1,N}$ – формируется датаграмма $\overline{Obj_k}$, включающая значение $MAC(Obj_k)$ (либо $TAG(Obj_k)$) и зашифрованные данные фрагмента Obj_k. Размеры полей данной датаграммы определяются используемыми в схеме алгоритмами блочного шифрования и имитозащиты. Таким образом, в облачном хранилище файл-контейнер Obj представляется последовательностью датаграмм Obj_k, $k = \overline{1,N}$. Структура такого файл-контейнера схематично представлена в табл. 2.

Размер последней датаграммы $\overline{Obj_N}$ может отличаться от размера остальных датаграмм $\overline{Obj_k}$, $k = \overline{1,N}$, поскольку объем данных, размещаемых в файл-контейнере Obj, необязательно кратен заданному размеру фрагмента и/или длине блока выбранного алгоритма блочного шифра (к таким данным применяется выравнивание в соответствии с алгоритмом, определенным в ГОСТ Р 34.12-2018).

Формат защищенных данных, размещаемых в облачном хранилище с использованием протокола CalDAV

Как уже отмечалось, протокол CalDAV предназначен для работы с информацией о календарях и расписаниях, представленной в виде объектов iCalendar. Каждый такой объект описывается набором свойств, являющихся, как правило, строго типизированными. Это накладывает серьезные ограничения на возможность применения к этим данным криптографических преобразований.

Пусть, например, пользователь размещает в календаре CalDAV информацию о некотором событии X, для чего передает в облачное хранилище объект iCalendar с компонентой VEVENT. Свойства последней, представленные в виде контентных строк, описывают событие X. Так, свойства SUMMARY,

Таблица 2.

Структура файл-контейнера с зашифрованными данными

Номер фрагмента	Тип данных	Длина в байтах
1	$MAC(Obj_1)$ или $TAG(Obj_1)$	$sizeof(MAC)$ или $sizeof(TAG)$
	Зашифрованные данные Obj_1	$sizeof(Obj_1)$
2	$MAC(Obj_2)$ или $TAG(Obj_2)$	$sizeof(MAC)$ или $sizeof(TAG)$
	Зашифрованные данные Obj_2	$sizeof(Obj_2)$
...	...	...
N	$MAC(Obj_N)$ или $TAG(Obj_N)$	$sizeof(MAC)$ или $sizeof(TAG)$
	Зашифрованные данные Obj_N	$sizeof(Obj_N)$

DESCRIPTION и LOCATION компоненты VEVENT содержат, соответственно, название, описание и место проведения события X, свойства DTSTART и DTEND содержат дату и время, соответственно, начала и завершения события X. В соответствии со спецификацией RFC 5545 первые три из перечисленных свойств хранят произвольные текстовые данные, а последние два свойства хранят данные типа DATE-TIME. Применяя классические криптографические преобразования (операции блочного шифрования) к значениям указанных свойств, мы получаем на выходе какие-то бинарные данные. Ясно, что такие данные являются допустимыми значениями для свойств SUMMARY, DESCRIPTION и LOCATION, поскольку всегда могут быть представлены в виде текста с помощью того или иного способа кодирования. В то же время очевидно, что бинарные данные, полученные в результате применения криптографических преобразований к данным типа DATE-TIME, не являются данными типа DATE-TIME и, более того, в общем случае не могут быть представлены (закодированы) как данные этого типа. Таким образом, криптографические методы защиты можно применять только к некоторой группе свойств объектов iCalendar, а для обеспечения информационной безопасности остальных нужно использовать другие (не криптографические) механизмы.

Авторы предлагают выполнять криптографические преобразования только тех значений свойств объектов iCalendar, которые хранят текстовые и бинарные данные, а в отношении других свойств поступать следующим образом: будем присваивать им «ложные» (не информативные), но допустимые с точки зрения формата значения, а подлинные (информативные) данные сохранять в зашифрованном виде в значениях свойств, к которым применялись криптографические преобразования. Так, например, при размещении в календаре CalDAV информации о новом событии, мы можем зашифровать значения свойств SUMMARY, DESCRIPTION и LOCATION компоненты календаря VEVENT, присвоить свойствам DTSTART и DTEND неактуальные значения даты и времени суток, тогда как подлинные значения в зашифрованном виде сохранить в значении свойства SUMMARY [2].

В соответствии с предлагаемой криптографической схемой параметры шифрования (в том числе экспортированные материалы мастер-ключей шифрования и имитозащиты) объекта iCalendar также размещаются в облачном хранилище для возможности многопользовательского доступа к зашифрованным данным. Отметим, что указанные ранее способы хранения такой информации в облаке неприменимы для CalDAV. Во-первых, CalDAV не предоставляет механизмы для создания в хранилище файл-контейнеров с произвольными пользовательскими данными. Во-вторых, CalDAV не позволяет определять собственные (не описанные в стандарте формата iCalendar) свойства для объектов iCalendar, которые можно было бы использовать для хранения служебной информации (аналогично тому, как параметры шифрования хранятся в метаданных файл-контейнеров WebDAV и S3).

В связи с этим авторы предлагают сохранять параметры шифрования объекта iCalendar в значении одного из его свойств, при этом такое значение удобно представить в виде JSON-токена с парами «ключ-значение», указанными в (табл. 1) (определяют параметры шифрования объекта iCalendar), и одним дополнительным элементом, содержащим зашифрованное значение свойства объекта iCalendar.

Условия и возможности применения постквантовых алгоритмов шифрования и(или) квантовозависимых ключей

Рассмотрим изменения криптографической схемы, предусматривающие использование квантовозависимых ключей шифрования из квантовых криптографических систем выработки и распределения ключей (ККС ВРК), а также технологические ограничения при использовании постквантовых алгоритмов шифрования.

Предлагаемая модификация криптографической схемы состоит в использовании ККС ВРК на этапе выработки и распределения «базовой» ключевой информации: ключа KEK_{VKO} (вариант 1) или мастер-ключей K_E, K_M (вариант 2).

Первый вариант предпочтителен с точки зрения скорости функционирования криптографической схемы в целом, так как выработка криптографически защищенных ключей в ККС осуществляется с невысокой скоростью относительно скорости шифрования. Такой вариант нивелирует и главную угрозу со стороны квантового вычислителя, предотвращая возможное раскрытие ключевой информации (секретных ключей x и y).

Второй вариант подразумевает выработку ключевой информации большего объема, но существенно упрощает криптографическую схему. При выработке ключей K_E, K_M с помощью ККС ВРК отпадает необходимость в шифровании, имитозащите этих ключей, а также в процедуре экспорта материала ключей. Таким образом, шаги 1–3 при размещении и получении данных заменяются одной процедурой:

Шаг 1’–3’. Совместная выработка мастер-ключей K_E, K_M в ККС ВРК.

Выбор конкретных параметров ККС ВРК влияет на скорость выработки КИ, предельно допустимое расстояние ее передачи, криптографическую стойкость ключей, возможность противодействия атакам со стороны потенциального нарушителя. Перечислим основные характерные черты возможной ККС ВРК:

реализация одно- или двухпроходной схемы передачи оптических сигналов; использование в оптической схеме одного или двух однофотонных детекторов; выбор сторонами номера оптического сигнала (в двухмодовом состоянии) для наложения фазового сдвига; выбор и, возможно, смена для разных сессий ВРК квантового протокола (BB84, SARG04, AKM2017, AKM2021, протоколов на симметричных когерентных состояниях) [11].

Одним из недостатков ККС является ограниченность расстояния передачи КИ в силу наличия оптических потерь. Выработка ключевой информации возможна на расстояниях от нескольких десятков до сотен километров, в зависимости от требований к *стойкости* вырабатываемых ключей. Эта проблема может быть решена, например, построением сетей ККС, обеспечивающих передачу КИ на (теоретически) неограниченные расстояния, и дающих возможность создания дублирующих линий передачи КИ для защиты от несанкционированного доступа.

Опыт разработки систем криптографической защиты информации на основе ККС ВРК[8,9] для организации защиты объектов критической информационной инфраструктуры позволил реализовать протоколы и интерфейсы взаимодействия КС ВРК и средств криптографической защиты информации, при этом разработанная квантовая криптографическая система обеспечила выработку криптостойкой ключевой информации со средней скоростью не менее 10 бит/с при дальности передачи квантовых сигналов не менее чем на 15 км, что достаточно для размещения в оптоволоконной инфраструктуре ЦОД. При этом взаимодействующие с ККС ВРК средства криптографической защиты информации могут обеспечивать обработку данных на скорости 10 Гбит/с и выше[10].

Рассмотрим изменения криптографической схемы, предусматривающие использование постквантовых алгоритмов; точнее – асимметричных криптосхем, способных противостоять угрозе использования квантовых алгоритмов и компьютеров. Заметим, что современный уровень развития квантовых вычислений позволяет считать существующие *симметричные* алгоритмы шифрования и функции хеширования (в том, числе использующие в СНГ: ГОСТ Р 34.10-2018, ГОСТ Р 34.11-2018), защищенными от квантовых компьютеров.

Наиболее актуальным с точки зрения рассматриваемых в данной статье задач является один из трех постквантовых федеральных стандартов обработки информации (Federal Information Processing Standard, FIPS), выпущенных в 2024 году институтом NIST: это стандарт FIPS 203, описывающий алгоритм инкапсуляции ключей (ML-KEM) на основе модульных решеток. В алгоритме ML-KEM[11] реализованы следующие процедуры:

- выработка открытого и закрытого ключей для, соответственно, инкапсуляции и декапсуляции общего ключа;
- выработка (инкапсуляция) общего ключа симметричного шифрования;
- восстановление (декапсуляция) общего ключа симметричного шифрования.

Отметим, что важным этапом алгоритма является вычисление шифртекста C (на этапе инкапсуляции) для последующей проверки декапсулированного ключа. В табл. 3 приведены значения длин (в байтах) шифртекста.

Для сравнения приведем параметры еще нескольких схем обмена ключевой информацией (табл. 4), (потенциально) имеющих высокий уровень стойкости[12,13].

Независимо от выбора варианта алгоритма, длина вырабатываемого ключа фиксирована и составляет

8 Ловков Д. А. Особенности применения квантового протокола выработки и распределения ключей Decoy States / Д.А. Ловков // Системы и средства защиты информации: Сборник докладов 11-й межведомственной научно-практической конференции, Пенза, 2019 / ПГУ. Пенза, 2020. 260 с. – Деп. в ООО «НТП Криптософт» 23.04.2024, № 8/24/пи.

9 Филиппов А. В. Вопросы построения квантовых каналов на основе промышленных волоконно-оптических линий связи // Системы и средства защиты информации: Сборник докладов 11-й межведомственной научно-практической конференции, Пенза, 2019. – Пенза: Изд-во ПГУ, 2020. 260 с. – Деп. в ООО «НТП Криптософт» 23.04.2024, № 8/24/пи.

10 А. А. Карманов. Способ и устройство квантового распределения ключей с контролем параметров квантового канала // Патент на изобретение RU 2840355 C1, 21.05.2025. Заявка № 2024117861 от 27.06.2024.

11 National Institute of Standards and Technology (2024) Module-Lattice-Based KeyEncapsulation Mechanism Standard. (Department of Commerce, Washington, D.C.), Federal Information Processing Standards Publication (FIPS) NIST FIPS 203. DOI: 10.6028/NIST.FIPS.203.

12 Гребнев С. В. (2019) Тенденции развития постквантовой криптографии. // Материалы XXI научно-практическая конференция «РусКрипто'2019». URL: https://ruscrypto.ru/accociation/archive/rc2019.html.

13 Панков К. Н., Миронов Ю. Б. Использование постквантовых алгоритмов в задачах защиты информации в телекоммуникационных системах. М.: Горячая линия, Телеком, 2023. 236 с.

Таблица 3.

Размеры шифртекста и ключей алгоритма ML-KEM

Вариант алгоритма	Длина ключа инкапсуляции	Длина ключа декапсуляции	Длина шифртекста
ML-KEM-512	800	1632	768
ML-KEM-768	1184	2400	1088
ML-KEM-1024	1568	3168	1568

Таблица 4.

Размеры параметров (в байтах) некоторых схем обмена ключами

Схема	Длина секретного ключа	Длина открытого ключа	Длина шифртекста
Three Bears	40	1584	1697
LEDAcrypt	40	18016	9008
FrodoKEM	31272	15632	15768
RQC	3510	3510	3574
SIKE	826	726	766

256 бит, что позволяет использовать ML-KEM для безопасной выработки ключевого материала в используемых алгоритмах блочного шифрования («Магма» и «Кузнечик»).

Практическое моделирование (реализация) метода и схемы криптографической защиты в протоколе доступа к облачному хранилищу

Представленные в данной работе механизмы криптографической защиты данных, передаваемых в облачные хранилища, смоделированы и проверены на практике[14] с использованием версий отечественных СКЗИ «QP Криптофон», «КриптоПро CSP», а также разработанного авторами экспериментального образца программного СКЗИ, реализующего алгоритмы блочного шифрования «Магма» и «Кузнечик» в стандартизированных режимах работы, определенных в ГОСТ 34.13-2018 и в перспективном режиме работы блочного шифра XEH [10].

Одной из задач экспериментального моделирования была оценка влияния различных режимов блочного шифра на скорость передачи данных в облачное хранилище. Для исследования этого аспекта была проведена серия испытаний, в которых измерялось время передачи (с шифрованием) данных разного размера в облачное хранилище «Яндекс.Диск» при помощи WebDAV. Для шифрования данных использовался алгоритм блочного шифра «Магма» в режимах работы CBC-MAC, CTR-ACPKM и XEH. Моделирование и испытания проводились на персональном компьютере, имитирующем автоматизированное рабочее место, подключённое по сети Интернет к указанному облачному хранилищу, с техническими характеристиками, представленными в (табл. 5).

На рис. 1 представлены графики зависимости времени передачи данных (в секундах) в облачное хранилище от объема передаваемых данных (в мегабайтах). Красная, зеленая и синяя кривые соответствуют передаче данных с шифрованием в режимах работы CBC-MAC, CTR-ACPKM и XEH; черная пунктирная линия соответствует передаче данных без шифрования. Для построения адекватных и объективных зависимостей каждая представленная на рис. 1 точка получена путем усреднения значений, полученных в десяти одинаковых измерениях.

На рис. 2 представлены графики зависимости коэффициента замедления передачи данных с шифрованием от объема передаваемых данных; красная кривая соответствует режиму CBC, зеленая кривая – CTR-ACPKM и синяя кривая – режиму XEH. Коэффициент замедления рассчитывается как частное от деления времени передачи данных с шифрованием на время передачи данных без шифрования. Точные значения коэффициента замедления представлены в табл. 6.

14 Минаков С. С., Тихов С. В. (2025) О механизмах криптографической защиты данных публичных облачных хранилищ и перспективах стандартизации технических спецификаций к прикладным протоколам облачных хранилищ данных // Материалы XXVII научно-практическая конференция «Рус-Крипто'2025». https://ruscrypto.ru/accociation/archive/ rc2025.html

Таблица 5.

Технические характеристики ПК

Процессор	Intel(R) Core(TM) i3-10100 CPU @ 3.60GHz
Объем ОЗУ	16 GB
Жесткий диск	HDD
Сетевой интерфейс	Ethernet 100 МБ/сек

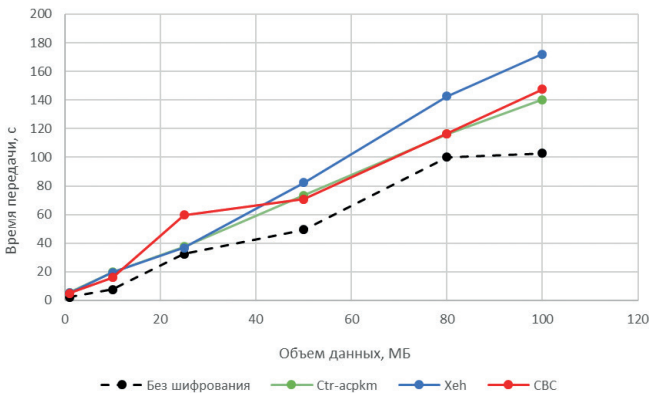

Рис. 1. Зависимость времени передачи данных в облачное хранилище от объема передаваемых данных

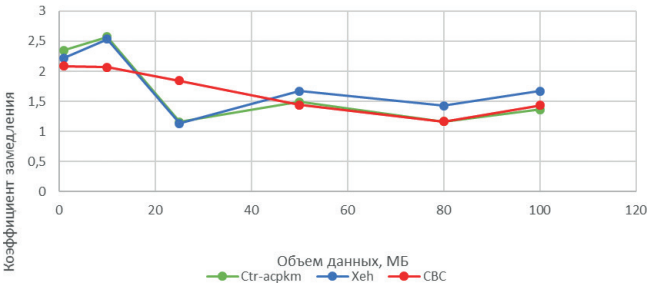

Рис. 2. Коэффициент замедления при передаче данных, обусловленный шифрованием данных, в зависимости от объема передаваемых данных

Представленные результаты показывают, что шифрование данных не приводит к серьезному замедлению системы передачи данных. Как следует из рис. 2, коэффициент замедления при размещении в облаке файлов размера менее 10 МБ равен приблизительно 2,5, а при размещении файлов размера более 25 МБ в среднем равен 1,3 для режимов работы блочного шифра CTR-ACPKM и CBC, и 1,47 для режима XEH, что является хорошим показателем для СКЗИ.

Таблица 6.

Коэффициент замедления при передаче данных с шифрованием в режимах CBC, CTR-ACPKM и XEH

Размер, МБ	CBC	CTR-ACPKM	XEH
1	2,09	2,35	2,22
10	2,06	2,57	2,53
25	1,84	1,16	1,13
50	1,44	1,49	1,67
80	1,16	1,16	1,42
100	1,43	1,36	1,67

Заключение

Развитие криптографической схемы класса «Утро», предложенное авторами, строится на композиции симметричных и асимметричных шифров [2] с криптоалгоритмами гарантированной стойкости. Такой метод защиты, как показано в работе предполагает возможность в будущем расширения квантовозависимыми ключами и новыми криптографическими алгоритмами постквантового класса.

В работе также показано, что метод сохранил инвариантность криптографической схемы «Утро-1» [1] относительно использования других алгоритмов шифрования, в том числе используемых в спецификациях интерфейса PKCS11 и для алгоритмов ГОСТ (СНГ) и ГОСТ Р, предлагается использовать параметры криптографических функций, опубликованных[15] Национальным технологическим центром цифровой криптографии или методическими рекомендациями[16] ТК 26 Росстандарта.

Нетрудно видеть, что аналогично можно реализовать метод защиты для формирования файлов-контейнеров в корпоративной сети с другими прикладными протоколами: CIFS (SMB), NFS в виде сервиса Network Attached Storage и на локально подключённых накопителях информации с различными файловыми системами.

Метод и схема криптографической защиты позволяют также строить и другие системы защищённого облачного хранения файлов для нескольких различных пользователей, для корпоративных доменов, например, на базе операционных систем Linux (AstraLinux, BaseALT и др.) и Microsoft Windows класса NT 5.xx (Windows 2000, XP, 2003, Windows FLP «Eiger»), NT 6.xx (Windows Vista, 7 — 11), в которых несложно построить процедуры поддержки форматной совместимости[17] для файлов-контейнеров EFS и сервиса Web Folder Environment с реализацией отечественной криптографической схемы в российских программных системах защиты информации[18,19].

15 «Расширение PKCS#11 для использования стандартов ГОСТ 34.12-2018, ГОСТ 34.13-2018, ГОСТ 34.10-2018, ГОСТ 34.11-2018» / Расширение спецификаций программного интерфейса PKCS#11 (версии 3.0 и выше). // М.: НТЦЦК, 2024 г. URL: https://digitalcryptography.ru/ upload/iblock /3d4/y8zw1jb7hhbelqf5qbypvo0hmm4yfl47.pdf.

16 МР 26.2.007-2017 «Расширение PKCS#11 для использования российских стандартов ГОСТ Р 34.10-2012 и ГОСТ Р 34-11-2012» (Документ утвержден решением заседания ТК 26, протокол № 20 от 24.11.2017) URL: https://tc26.ru/standarts/metodicheskie-rekomendatsii/mr-26-2-007-2017-rasshirenie-pkcs-11-dlya-ispolzovaniya-rossiyskikh-standartov-gost-r-34-10-2012-i-gost-r-34-11-2012.html.

17 How Encrypting File System Works: Security Policy; Public Key; Security Services // URL: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc781588(v=ws.10).

18 ПСЗИ семейства QP // URL: https://cryptosoft.ru/progr.html

19 Secret Net Studio 8.10 // URL: https://www.securitycode.ru/products/secret-net-studio

Литература

1. Минаков С. С. Основные криптографические механизмы защиты данных, передаваемых в облачные сервисы и сети хранения данных // Вопросы кибербезопасности. 2020. № 3(37). С. 66–75. DOI: 10.21681/2311-3456-2020-03-66-75.
2. Вариант построения программного решения с гибридной криптографической системой защиты данных, хранящихся на облачном накопителе, и перспективными режимами работы блочных шифров / С. С. Минаков, И. В. Карпов, С. В. Тихов, И. В. Мартынов // Системы и средства защиты информации: Сборник статей 16-й межведомственной научно-практической конференции имени Е. А. Матвеева, Пенза, 10–13 сентября 2024 года. – Пенза: Изд-во ПГУ. 2025. С. 94–105.
3. Quantum Computers as Universal Quantum Simulators: State-of-the-Art and Perspectives / F. Tacchino, A. Chiesa, S. Carretta, D. Gerace // Adv. Quantum Technol. 2020. № 3. 1900052. DOI: 10.1002/qute.201900052.

4. Ball P. Physicists in China challenge Google's 'quantum advantage' // Nature. 2020. № 588(380). DOI: 10.1038/d41586-020-03434-7.
5. Wiring surface loss of a superconducting transmon qubit / N. S. Smirnov, E. A. Krivko, A. A. Solovyova [et al.] // Sci. Rep. 2024. V. 14. P. 7326. DOI: 10.1038/s41598-024-57248-y.
6. Минаков С. С. Актуальные научные вопросы осуществления технико-криминалистических мероприятий и применения инструментальных средств при реагировании на компьютерные инциденты и анализе распределенных защищенных систем, обрабатывающих сведения конфиденциального характера // Судебная экспертиза и исследования. 2024. № 4. С. 123–129.
7. Зубов А. Ю. Криптосистема блочного гаммирования с аутентификацией // Математические вопросы криптографии. 2022. № 4(13). С. 5–35.
8. Защищённое хранение данных и полнодисковое шифрование / Е. К. Алексеев, Л. Р. Ахметзянова, А. А. Бабуева, С. В. Смышляев // Прикладная дискретная математика. 2020. № 49. С. 78–97. DOI: 10.17223/20710410/49/6.
9. Bogdanov D. S., Nozdrunov V. I. Some properties of the DEC mode of operation of block ciphers // Математические вопросы криптографии. 2022. № 3(13). С. 37–44.
10. Коренева А. М., Фирсов Г. В. Об одном режиме работы блочных шифров для защиты системных носителей с блочно-ориентированной структурой // Прикладная дискретная математика. Приложение. 2023. № 16. С. 52–56. DOI: 10.17223/2226308X/16/14.
11. Класс квантовых криптографических систем АКМ2021 на основе использования синглетных состояний многокубитовых квантовых систем / Ф. К. Алиев, А. В. Корольков, Е. А. Матвеев // Системы высокой доступности. 2022. № 3(18). С. 5–22.

.

THE CRYPTOGRAPHIC PROTECTION SCHEME OF CLOUD STORAGE DATA WITH POSSIBILITY TO EXPANDING BY QUANTUM-DEPENDENT KEYS AND POST-QUANTUM ENCRYPTION ALGORITHMS

Minakov S. S.[20], Tikhov S. V.[21], Tsupak A. A.[22]

Keywords: *encryption, cloud storage, cryptographic security, network access, application-level protocol, information security, key generation.*

Purpose of the study: *the paper focuses on the development of scheme for cryptographic protection data in cloud storages using standardized and emerging block-cipher modes, post-quantum encryption algorithms, and quantum-dependent keys.*

Methods of research: *system-level analysis of information security threats during processing and storage when using application-layer access protocols for cloud storage. Synthesis of cryptographic mechanisms and formats to build an automated hybrid cryptographic scheme for client-side information processing using symmetric and asymmetric ciphers.*

Result(s): *the article presents the development of the Utro-1 cryptographic scheme for ensuring information security in cloud storage access application protocols. Formats, functions and encryption logic are described; several practical tests of the software implementation are carried out. The article provides explanations on the conditions for using post-quantum algorithms and/or quantum-dependent keys in the scheme.*

Scientific novelty: *the scientific novelty and relevance lie in achieving new scientific and technical results in the field of protecting data transmitted to cloud storage using Russian and promising foreign cryptographic tools and methods; in developing a cryptographic scheme to protect information from unauthorized access by means of a hybrid composition of symmetric and asymmetric ciphers.*

References

1. Minakov, S. S. (2020). The main cryptographic mechanisms for protection of data, transmitted to cloud services and storage area networks. Cybersecurity issues., 3(37), 66–75. DOI 10.21681/2311-3456-2020-03-66-75.
2. Minakov, S. S., Karpov, I. V., Tikhov, S. V., Martynov, I. V. (2025). Variant postroeniya programmnogo resheniya s gibridnoj kriptograficheskoj sistemoj zashhity dannyx, xranyashhixsya na oblachnom nakopitele, i perspektivnymi rezhimami raboty blochnyx shifrov. Sistemy i sredstva zashhity informacii: Sbornik statej 16-j mezhvedomstvennoj nauchno-prakticheskoj konferencii imeni E. A. Matveeva, Penza, 10–13 sentyabrya 2024 goda, 94–105.
3. Tacchino, F., Chiesa, A., Carretta, S. and Gerace, D. (2020), Quantum Computers as Universal Quantum Simulators: State-of-the-Art and Perspectives. Adv. Quantum Technol., 3, 1900052. DOI 10.1002/qute.201900052.
4. Philip Ball. (2020). Physicists in China challenge Google's 'quantum advantage'. Nature., 588, 380. DOI 10.1038/d41586-020-03434-7.
5. Smirnov, N. S., Krivko, E. A., Solovyova, A. A. et al. (2024). Wiring surface loss of a superconducting transmon qubit, Sci Rep., 14, 7326. DOI 10.1038/s41598-024-57248-y.

20 Sergey S. Minakov, Senior research scientist (S. R. O.), Russian Academy of Cryptography. Moscow, Russia. E-mail: ss_minakov@mail.ru
21 Stanislav V. Tikhov, Senior developer, ООО NTP «Cryptosoft». Penza, Russia. E-mail: tik.stanislav2015@yandex.ru
22 Aleksei A. Tsupak, Dr.Sc., Associate Professor, Penza State University. Penza, Russia. E-mail: altsupak@yandex.ru

6. Minakov, S. S. (2024). Aktual'nye nauchnye voprosy osushhestvleniya texniko-kriminalisticheskix meropriyatij i primeneniya instrumental'nyx sredstv pri reagirovanii na komp'yuternye incidenty i analize raspredelennyx zashhishhennyx sistem, obrabatyvayushhix svedeniya konfidencial'nogo xaraktera. Sudebnaya ekspertiza i issledovaniya., 4, 123–129.
7. Zubov, A. Yu. (2022). Kriptosistema blochnogo gammirovaniya s autentifikaciej. Mathematical Aspects of Cryptography., 13(4), 5–35.
8. Alekseev, E. K., Akhmetzyanova, L. R. Babueva, A. A. Smyshlyaev, S. V. (2020). Data storage security and full disk encryption. PDM., 49, 78–97. DOI 10.17223/20710410/49/6.
9. Bogdanov, D. S., Nozdrunov, V. I. (2022). Some properties of the DEC mode of operation of block ciphers, Mathematical Aspects of Cryptography, 13(3), 37–44.
10. Koreneva, A. M., Firsov, G. V. (2023). Ob odnom rezhime raboty blochnyx shifrov dlya zashhity sistemnyx nositelej s blochno-orientirovannoj strukturoj. PDM Prilozhenie. 16, 52–56, DOI 10.17223/2226308X/16/14.
11. Aliev, F. K., Korolkov, A. V., Matveev, E. A. (2022). Class of quantum cryptographic systems AKM2021 based on the use of singlet states of multicubic quantum systems // Journal Highly available systems, 18(3), 5–22.