УДК 004.89

ОЦЕНКА НАДЕЖНОСТИ МЕТОДОВ СИММЕТРИЧНОГО И АСИММЕТРИЧНОГО ШИФРОВАНИЯ ДАННЫХ И МЕТОДА ХЕШИРОВАНИЯ

[1]Паршин И.О., [2]Чураев В.С.

[1]Федеральное государственное унитарное предприятие «Всероссийский научно-исследовательский институт «Центр», 123242, г. Москва, ул. Садовая — Кудринская, д. 11, строение 1, e-mail: centr@vniicentr.ru

[2]МИРЭА - Российский технологический университет, 119454, Россия, г. Москва, проспект Вернадского, 78, e-mail: vyacheslav.churaev.1901@yandex.ru

В статье рассматриваются различные методы шифрования данных для обеспечения безопасности. Были проведены эксперименты, сравнивающие эффективность трех методов: симметричное шифрование, асимметричное шифрование и хеширование. Для оценки надежности каждого метода использовались метрики безопасности, такие как криптостойкость, скорость шифрования и дешифрования, а также устойчивость к атакам.

Ключевые слова: криптография, симметричное шифрование, ассиметричное шифрование, хэширование.

ASSESSMENT OF THE RELIABILITY OF SYMMETRIC AND ASYMMETRIC DATA ENCRYPTION METHODS AND HASHING METHOD

[1]Parshin I.O., [2]Churaev V.S.

[1]Federal State Unitary Enterprise "All-Russian Scientific Research Institute "Center", 123242, Moscow, Sadovaya — Kudrinskaya str., 11, building 1, e-mail: centr@vniicentr.ru

[2]MIREA - Russian Technological University, 119454, Moscow, 78 Vernadsky Avenue, Russia, e-mail: vyacheslav.churaev.1901@yandex.ru

The article discusses various methods of data encryption aimed at ensuring security. Experiments were conducted to compare the effectiveness of three methods: symmetric encryption, asymmetric encryption, and hashing. Security metrics such as cryptographic strength, encryption and decryption speed, and resistance to attacks were utilized to evaluate the reliability of each method.

Keywords: cryptography, symmetric encryption, asymmetric encryption, hashing.

Введение

В прошлом шифрование использовалось государственными органами для передачи секретной информации в процессе коммуникации. Однако в современном мире шифрование широко применяется в различных сферах, особенно при передаче данных через сети, устройства Bluetooth, банкоматы и т. д. Шифрование играет важную роль в обеспечении безопасности информации, которую часто сложно физически защитить. Все большее число людей используют компьютерные приложения для различных задач, включая покупки и продажи онлайн, общение с родственниками за границей и передачу конфиденциальных данных (таких как информация о кредитных картах, банковских счетах и личной переписке) через Интернет. В таком контексте использование шифрования становится неотъемлемой частью обеспечения безопасности и конфиденциальности в онлайн-среде.

Шифром называют систему или алгоритм, трансформирующий произвольное сообщение в такую форму, которую не сможет прочитать никто кроме тех, кому это сообщение предназначено. Сущность шифрования состоит в задании соответствия между произвольным исходным сообщением и зашифрованным результирующим текстом, при этом должен существовать способ задать обратное соответствие, при помощи которого можно восстановить исходный текст [1].

Существует множество методов шифрования данных, каждый из которых имеет свои достоинства и недостатки. Все методы можно разделить на группы в зависимости от количества ключей, которые используются в соответствующих алгоритмах: двухключевые; одноключевые; бесключевые.

В двухключевых алгоритмах используется два ключа: открытый и закрытый. В одноключевых используется обычный секретный ключ. В бесключевом, соответственно, не используются ключи. Кроме разделения методов

шифрования на группы, их также можно разделить на категории:

алгоритмы ассиметричного шифрования;

алгоритмы симметричного шифрования;

хеширование.

Алгоритмы симметричного шифрования используют один и тот же ключ для обоих операций – шифрования и расшифрования данных. Ключ является секретным и должен быть передан от отправителя к получателю безопасным способом. Симметричные шифровальные алгоритмы должны полностью устранять статистические закономерности и не должны быть линейными.

В таких алгоритмах можно выделить две основные категории: блочные и потоковые. В блочных системах данные делятся на блоки, и затем каждый блок преобразуется с использованием ключа. В потоковых системах генерируется последовательность данных (гамма), которая применяется к исходным данным для их шифрования. Схема связи с использованием симметричной криптосистемы представлена на рисунке 1.

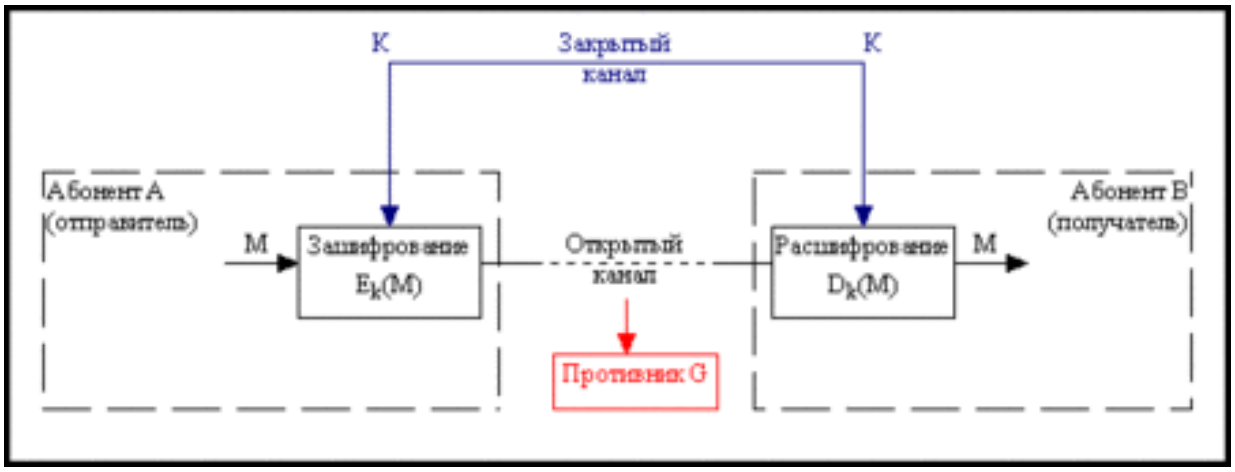

Рисунок 1. Схема симметричного шифртекстования, где M - открытый текст, K - секретный ключ, передаваемый по закрытому каналу, En(M) - операция зашифрования, Dk(M) - операция расшифрования

При симметричном шифровании обычно используется сложная и многоступенчатая комбинация подстановок и перестановок исходных данных. Каждая ступень или проход требует своего «ключа прохода». Операция подстановки выполняет первое требование симметричного шифра путем перемешивания битов сообщения в соответствии с заданным законом, что устраняет любые статистические данные. Перестановка необходима для второго требования – придания алгоритму нелинейности. Это достигается заменой определенной части сообщения заданного объема на стандартное значение, основываясь на исходном массиве данных. Симметричные системы имеют свои преимущества и недостатки по сравнению с асимметричными. Основные преимущества симметричных шифров включают:

высокую скорость шифрования;

меньшую длину ключа при сохранении стойкости;

лучшую изученность и простоту реализации.

Недостатками симметричных алгоритмов являются:

сложность обмена ключами, так как существует вероятность нарушения секретности ключа при обмене;

сложность управления ключами в больших сетях.

Кроме достоинств и недостатков, каждый метод шифрования имеет свои метрики безопасности, такие как криптостойкость, скорость шифрования и дешифрования, устойчивость. Криптостойкость является ключевой характеристикой шифровальных алгоритмов и обычно измеряется временем, которое злоумышленнику потребуется для вскрытия шифра при определенных ресурсах, которыми он располагает [2, 3].

Существует классификация атак на симметричные алгоритмы шифрования, которая включает следующие типы атак.

1. Атака с известным открытым текстом: криптоаналитик имеет доступ к парам текстов, состоящих из открытого текста и соответствующего шифртекста.

2. Атака с выбранным открытым текстом: криптоаналитик может выбирать открытые тексты и получать соответствующие шифртексты.

3. Адаптивная атака с выбором открытого текста: криптоаналитик может не только выбирать открытые тексты для шифрования, но и использовать результаты анализа предыдущих данных и повторять процесс многократно.

4. Атака с выбором шифртекста: криптоаналитик может выбирать шифртексты и получать соответствующие открытые тексты.

5. Адаптивная атака с выбором шифртекста: криптоаналитик может многократно выбирать шифртексты для их расшифрования, учитывая предыдущие результаты анализа.

Рассмотрим криптоаналитические методы, которые используются в атаках.

Метод «грубой силы» предполагает перебор всех возможных комбинаций ключа шифрования для поиска искомого ключа. Защита от атак данного метода проста: увеличение размера ключа на 1 бит увеличит возможное количество ключей, что усложнит атаку методом «грубой силы». Метод может быть использован в сочетании с другими методами криптоанализа.

Метод «встречи посередине» используется для вскрытия алгоритмов шифрования, использующих двойное шифрование. Например, атака может быть направлена на алгоритм Double DES [4].

Дифференциальный криптоанализ базируется на анализе пар открытых текстов с определенной разностью. С помощью этого метода можно вскрыть однораундовый DES и сократить количество комбинаций для трехраундового DES.

Линейный криптоанализ находит соотношения между открытым текстом, шифртекстом и ключом. Метод ищет однораундовые соотношения и пытается их распространить на большее количество раундов. Применяется к алгоритмам DES, RC5, NUSH и Noekeon.

Метод бумеранга является усилением дифференциального криптоанализа путем использования четырех вместо двух открытых текстов. Он представляет собой атаку с адаптивным выбором открытых текстов, и шифртекстов, которая на практике сложно применима. Он был применен против CAST-256, MARS и SERPENT.

Сдвиговая атака успешна независимо от количества раундов атакуемого алгоритма. Например, TREYFER был полностью вскрыт с помощью этой атаки.

Метод также применим к модифицированным версиям DES и Blowfish. Однако с помощью данной атаки можно вскрыть только те алгоритмы, раунды которых являются идентичными.

Метод интерполяции применим к алгоритмам, использующим простые алгебраические операции, позволяя криптоаналитику построить полином, который связывает шифртекст с открытым текстом.

Невозможные дифференциалы используют дифференциалы с нулевой или минимальной вероятностью с целью сократить множество возможных ключей для дальнейшего перебора и поиска секретного ключа. Этот метод нашел применение для вскрытия усеченных версий блочных алгоритмов шифрования, таких как IDEA [1, 5].

В данной статье рассматриваются блочные алгоритмы шифрования, такие как: AES, RC6, SERPENT, Twofish, поскольку они соответствуют определенным критериям для тестирования.

Размер блока шифруемых данных должен быть 128 бит.

Алгоритм должен поддерживать ключи шифрования размерами 128, 192 и 256 бит.

Алгоритм должен быть устойчивым против современных криптоаналитических атак.

Структура и математическая модель алгоритма должны быть понятными и простыми для облегчения изучения системы шифрования.

Алгоритм должен не иметь слабых или эквивалентных ключей.

Скорость шифрования должна быть высокой.

Алгоритм должен требовать минимального объема оперативной памяти.

Множество симметричных систем шифрования были исключены из анализа по различным причинам, таким как сложные математическая модель и операции, недостаточная стойкость против криптоаналитических атак, низкая скорость шифрования и невозможность реализации на разных платформах. Изучались следующие компоненты каждого алгоритма: структурная сложность метода шифрования и возможность расширения секретного ключа до размеров 128, 192 и 256 бит.

В таблице 1 приводятся результаты сравнительного анализа эффективности этих криптосистем по указанным критериям. Для анализа данных криптосистем использовался следующий подход: если критерий полностью не реализуется в определенном алгоритме, ему присваивался количественный показатель равный 0. Если критерий частично реализуется, ему присваивалась количественная оценка 0,5.

Следовательно, если критерий полностью реализуется в методе шифрования без каких-либо ограничений, ему присваивался показатель 1.

Таблица 1. Сравнительный анализ эффективности симметричных блочных криптосистем

Критерий	AES	RC6	SERPENT	Twofish
Криптостойкость	1	1	1	1
Запас криптостойкости	1	1	1	1
Скорость расширения ключа	1	0.5	0.5	0
Защита от атак по времени выполнения	1	0	1	0.5
Реализация лавинного эффекта	1	1	1	1
Возможность быстрого расширения ключа	0.5	0.5	1	1
Возможность параллельных вычислений	1	0.5	0.5	0.5
Результат	6.5	4.5	6	5

Изучение различных симметричных шифров показало, что криптостойкость данных методов является достаточной. Множество литературных источников подтверждают, что эти алгоритмы сложны для криптоаналитических атак, будь то полные или усеченные версии алгоритмов.

Запас стойкости определяется как соотношение между общим количеством раундов и наиболее уязвимым вариантом, подверженным криптоаналитическим атакам. Например, алгоритм SERPENT, состоящий из 32 раундов, может быть вскрыт с использованием дифференциально-линейного криптоанализа всего в 11 раундов.

Защита от атак по времени выполнения заключается в контроле скорости шифрования или расширения ключа. Если эти операции занимают неоправданно большое время, то алгоритм может быть уязвим к таким атакам.

Все рассмотренные симметричные шифры полностью реализуют лавинный эффект, поэтому получают оценку 1 по данному критерию.

Исследования показали, что все рассмотренные симметричные шифры поддерживают возможность быстрого расширения ключа, однако лишь SERPENT и Twofish реализуют эту возможность без ограничений.

Алгоритм AES является наиболее стойким среди всех рассмотренных и получает оценку 6,5 по шкале от 0 до 7 [6, 6].

В асимметричном шифровании используется пара ключей – публичный ключ для шифрования данных и приватный ключ для их расшифровки. При этом публичный ключ может быть распространен открыто, тогда как приватный ключ должен быть храниться в секрете. Приватный ключ не может быть вычислен на основе публичного ключа, так что обратное вычисление невозможно без знания приватного ключа. Схема передачи данных между двумя субъектами (А и Б) с использованием открытого ключа выглядит следующим образом.

Субъект А генерирует пару ключей, открытый и закрытый (публичный и приватный).

Субъект А передает открытый ключ субъекту Б. Передача может осуществляться по незащищенным каналам.

Субъект Б шифрует пакет данных при помощи полученного открытого ключа и передает его А. Передача может осуществляться по незащищенным каналам.

Субъект А расшифровывает полученную от Б информацию при помощи секретного, закрытого ключа.

В такой схеме перехват любых данных, передаваемых по незащищенным каналам, не имеет смысла, поскольку восстановить исходную информацию возможно только при помощи закрытого ключа, известного лишь получателю и не требующего передачи.

Преимущества асимметричного шифрования.

1. Безопасный обмен ключами: асимметричное шифрование позволяет безопасно передавать ключи и устанавливать общие секреты между двумя или более сторонами, даже через незащищенные каналы связи.
2. Шифрование и подпись данных: асимметричное шифрование позволяет шифровать данные и подписывать их с помощью приватного ключа, обеспечивая аутентификацию и целостность информации.
3. Открытый ключ не является секретным: для использования асимметричного шифрования не требуется секретное распространение открытого ключа, что облегчает процесс обмена ключами и управления ключевой инфраструктурой.
4. Удобство ключевого обмена: асимметричное шифрование и протоколы, такие как Диффи-Хеллман, способствуют безопасному обмену ключами между двумя сторонами без необходимости предварительного согласования ключевых материалов.
5. Распределенное шифрование: с помощью асимметричного шифрования можно реализовать защиту информации в распределенных системах, где каждый участник имеет свой собственный ключ.

Недостатки асимметричного шифрования.

1. Высокая вычислительная сложность: алгоритмы асимметричного шифрования требуют большего количества вычислительных ресурсов по сравнению с симметричными алгоритмами. Это может замедлить процессы шифрования и расшифрования данных.
2. Ограниченная длина сообщения: в некоторых алгоритмах асимметричного шифрования ограничена длина сообщения, которую можно зашифровать. Для шифрования более длинных сообщений может потребоваться использование гибридных схем, включающих симметричное шифрование.
3. Зависимость от инфраструктуры открытых ключей: асимметричное шифрование требует надежной инфраструктуры открытых ключей (PKI) для подтверждения подлинности открытых ключей. Управление PKI может быть сложной задачей.
4. Уязвимость к атакам с использованием квантовых компьютеров: некоторые алгоритмы асимметричного шифрования, такие как RSA и Эль-Гамаля, могут быть уязвимы к атакам с использованием квантовых компьютеров, которые могут быстро факторизовать большие числа и решить задачу дискретного логарифмирования.

Примерами алгоритмов асимметричного шифрования являются алгоритмы шифрования RSA, Диффи Хеллмана, Рабина и Эль-Гамаля.

Схема обмена ключами Диффи-Хеллмана (DH) представляет собой криптографический протокол, позволяющий двум или более сторонам получить общий секретный ключ даже при использовании незащищенного канала связи. Основной принцип безопасности протокола DH заключается в сложности задачи дискретного логарифмирования.

Алгоритм шифрования в криптосистеме Рабина также эффективен и подходит для использования на портативных устройствах. Он быстрее, чем алгоритм RSA, так как не требует возведения в степень, а использует возведение в квадрат. Однако у Рабина есть некоторые недостатки, включая неоднозначность расшифрования и уязвимость, связанная с использованием только квадратных остатков.

Схема Эль-Гамаля (ElGamal) является вариантом алгоритма DH и также использует открытый ключ. Оба абонента в этой схеме договариваются о общем секретном ключе, а затем сообщение шифруется путем умножения его на секретный ключ. Процесс шифрования включает генерацию ключей, алгоритм шифрования и алгоритм дешифрования. Шифрование Эль-Гамаля вероятностное, что означает, что один и тот же открытый текст может быть зашифрован несколькими разными шифртекстами, в результате чего размер шифртекста увеличивается в два раза по сравнению с исходным текстом. Это один из недостатков данного алгоритма. Однако шифрование и расшифровка требуют только нескольких операций возведения в степень, которые могут быть вычислены заранее. Криптостойкость алгоритма Эль-Гамаля основана на сложности задачи дискретного логарифмирования. При одинаковой длине ключа криптостойкость Эль-Гамаля сравнима с RSA, однако она менее популярна и требует более сложной реализации.

На основе приведённых данных осуществлена сравнительная характеристика алгоритмов шифрования в таблице 2. В качестве шести сравниваемых характеристик были выбраны те же характеристики, что и в таблице 1. Сравнение характеристик проводились при равной длине исходного сообщения m и длине ключе 4096 бит.

Изучение различных асимметричных шифров показало, что криптостойкость этих методов является достаточной. Множество литературных источников подтверждают, что эти алгоритмы сложны для криптоаналитических атак, будь то полные или усеченные версии алгоритмов. Из таблицы 2 можно увидеть, что наиболее устойчивым к атакам является метод Эль-Гамаля [7, 8].

При хешировании используется алгоритм, который преобразует входные данные произвольной длины в фиксированную строку (хеш). Хеш-функция обладает следующими свойствами: для одних и тех же входных данных всегда будет генерироваться один и тот же хеш, даже незначительное изменение входных данных должно привести к полностью различающимся хешам, возможность получения исходных данных на основе хеша должна быть вычислительно неосуществимой. В настоящее время в MD5 и вариантах SHA очень часто используются хэш-алгоритмы.

Хэш-функции — это математические вычисления, которые принимают произвольный объем данных в качестве входных данных и выдают выходные данные фиксированного размера. Если для одного и того же входного сигнала всегда получается один и тот же результат, это означает, что функция является детерминированной. В случае хэш-функций входные данные, которые называются сообщениями, преобразуются в выходные данные, известные как дайджесты сообщений. Хэш-функции широко используются для обеспечения целостности и аутентичности данных. Если дайджест сообщения изменяется, это означает, что само сообщение изменилось, что может служить индикатором изменений в файле или данных. Как примеры использования сообщений могут выступать текстовые строки, двоичные файлы или TCP-пакеты.

Таблица 2. Сравнительный анализ эффективности асимметричных криптосистем

Критерий	RSA	Диффи-Хеллмана	Рабина	Эль-Гамаля
Криптостойкость	1	1	1	1
Запас криптостойкости	1	1	0.5	1
Скорость расширения ключа	0.5	0.5	1	1
Защита от атак по времени выполнения	1	1	0.5	1
Реализация лавинного эффекта	0.5	0	0	0.5
Возможность быстрого расширения ключа	0.5	1	1	1
Возможность параллельных вычислений	0.5	1	0.5	0.5
Результат	5	5.5	4.5	6

Односторонняя хэш-функция (OWHF), созданная Мерклом, является функцией H, которая обладает следующими свойствами.

1. H может быть применена к блокам данных произвольной длины.
2. H генерирует выходные данные фиксированной длины, то есть дайджест сообщения.
3. При заданной функции H и входных данных x легко можно вычислить дайджест сообщения H(x).
4. При заданной функции H и дайджесте сообщения H(x) вычислительно невозможно найти исходные данные x.
5. При заданной функции H и дайджестах сообщений H(x) и H(x') вычислительно невозможно найти две различные исходные данные x и x', такие как H(x) = H(x').

Алгоритм MD5 представляет собой широко используемую хэш-функцию, которая генерирует 128-битное хэш-значение. Несмотря на то, что MD5 был изначально разработан как криптографическая хэш-функция, он показал значительные уязвимости.

Однако MD5 все еще может быть применен для некриптографических целей, например, для разделения данных в базе данных. Алгоритм MD5 принимает входное сообщение и разбивает его на фрагменты размером 512 бит. Затем сообщение дополняется таким образом, чтобы его общая длина стала кратной 512 битам. В конце дополнения добавляются нули, чтобы достичь длины сообщения, меньшей чем кратное 512 битам. Последние 64 бита состоят из значения, представляющего длину исходного сообщения по модулю 2^64.

Основной алгоритм MD5 работает с 128-битным состоянием, разделенным на четыре 32-битных слова — A, B, C и D. Эти слова инициализируются фиксированными константами. Затем алгоритм обрабатывает каждый 512-битный блок сообщения по очереди, изменяя состояние. Обработка блока сообщения состоит из четырех раундов, каждый из которых состоит из 16 операций на основе нелинейной функции F, модульного сложения и циклического сдвига влево.

В результате выполнения алгоритма MD5 получается 128-битное хэш-значение, которое можно использовать для целей проверки целостности данных. Однако из-за обнаруженных уязвимостей рекомендуется использовать более современные и надежные криптографические хэш-функции, такие как SHA-256.

SHA-1 (Secure Hash Algorithm) — это криптографическая хэш-функция, которая принимает данные на вход и выдает 160-битное (20-байтовое) хэш-значение, известное как дайджест сообщения. SHA-1 был широко используемым веб-браузерами для SSL-сертификатов, но с 2017 года все основные поставщики веб-браузеров прекратили принимать SSL-сертификаты, основанные на SHA-1. SHA-1 все еще используется в нескольких широко распространенных приложениях и протоколах безопасности, таких как TLS и SSL, PGP, SSH, S/MIME и IPSec для проверки целостности передаваемых данных.

Он позволяет определить, были ли внесены какие-либо изменения в информацию во время ее передачи от отправителя к получателю.

Однако из-за обнаруженных коллизий, то есть возможности нахождения двух разных сообщений с одинаковым хэш-значением, SHA-1 потерял свою надежность в криптографических приложениях. Рекомендуется использовать более сильные и безопасные хэш-функции, такие как SHA-256 для обеспечения достаточного уровня безопасности в криптографии.

Сравнение схем вычисления алгоритмов представлено на рисунке 2.

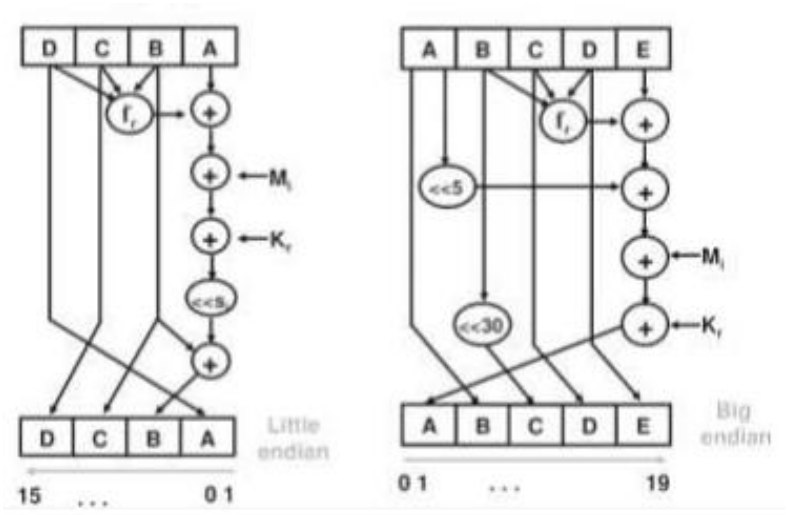

Рисунок 2. Сравнение работы алгоритмов MD5 и SHA-1

Сравнение алгоритмов хеширования будет производиться по следующим критериям: размер блока (в битах), размер дайджеста (выхода в битах), размер слова (в битах), количество раундов, встреча коллизий, поддерживаемые операции. Данные представлены в таблице 3.

Таблица 3. Сравнение популярных алгоритмов хеширования

Параметры	MD5	SHA-1	SHA-2	SHA-3	Whirpool
Размер блока (биты)	512	512	512,1024	1600–2*bits	512
Размер дайджеста (биты)	128	160	160,224,256, 384,512	160,224,256, 384,512	512
Размер слов (биты)	32	32	32,64	64	8
Раундов	4	80	80	24	10
Нахождение коллизий	Да	Да	Нет	Нет	Да
Поддерживаемые операции	And, or, xor, not	And, or, xor, not	And, or, xor, not, shr	And, or, xor, not, shr	And, or, xor, not

По данным таблицы можно сделать вывод, что наиболее предпочтительными алгоритмами являются: SHA-2 и SHA-3 [1,2].

Заключение

1. Рассмотрены симметричные и ассиметричные методы шифрования, а также алгоритмы хеширования.
2. Подробно описаны каждый метод и алгоритм, их достоинства и недостатки.
3. Проведен сравнительный анализ методов симметричного и ассиметричного шифрования, а также алгоритмов хеширования. Наиболее предпочтительными методами являются AES в симметричном шифровании, Эль-Гамаля в ассиметричном и SHA-2, SHA – 3 в алгоритмах хеширования.

Список литературы

1. Торстейнсон П., Ганеш Г.А. Криптография и безопасность в технологии .NET. Перевод с англ. – 4-е изд., электрон. – М.: Лаборатория знаний, 2020. – 482 с.
2. Фомичев В.М. Криптографические методы защиты информации. Курс лекций. – М.: Прометей, 2023. – 340 с.
3. Башир И. Блокчейн: архитектура, криптовалюты, инструменты разработки, смарт-контракты. Перевод с англ. М.А. Райтмана. – М.: ДМК Пресс, 2019. – 538 с.
4. Бабенко Л.К., Ищукова Е.А. Современные алгоритмы блочного шифрования и методы их анализа. – М.: Гелиос АРВ, 2006. –376 с.
5. Урбанович П.П. Защита информации методами криптографии, стеганографии и обфускации. – Минск:

БГТУ, 2016. – 220 с.

6. Панасенко С.П. Алгоритмы шифрования. Специальный справочник. – СПб.:БХВ-Петербург, 2009. 576 с.

7. Бабаш А.В., Баранова Е.К. Криптографические методы и средства защиты информации. Учебник. – М.: Кнорус, 2024. – 224 с.

8. Нестеров С.А. Основы информационной безопасности. Учебник – 2-е изд. – Санкт-Петербург: Лань, 2022. – 324 с.

References

1. Torsteinson, P., Ganesh, G.A. Cryptography and Security in .NET Technology. Translated from English - 4th edition electron. – Moscow: Knowledge Laboratory, 2020. – 482 p.

2. Fomichev V.M. Cryptographic Methods of Information Protection. Lecture Course. – Moscow: Prometheus, 2023. – 340 p.

3. Bashir I. Blockchain: Architecture, Cryptocurrencies, Development Tools, Smart Contracts. Translated from English by M.A. Wrightman. – M.: DMK Press, 2019. – 538 p.

4. Babenko L.K., Ishchukova E.A. Modern Block Cipher Algorithms and Methods of their Analysis. – Moscow: Helios ARV, 2006. – 376 p.

5. Urbanovich P.P. Information Protection by Cryptography, Steganography, and Obfuscation Methods. – Minsk: BSTU, 2016. – 220 p.

6. Panasenko S.P. Encryption Algorithms. Special Handbook. – St. Petersburg: BHV-Petersburg, 2009. – 576 p.

7. Babash A.V., Baranova E.K. Cryptographic Methods and Means of Information Protection. Textbook. – Moscow: Knorus, 2024. – 224 p.

8. Nesterov S.A. Fundamentals of Information Security. Textbook – 2nd ed. – St. Petersburg: Lan, 2022 – 324 p.