УДК 511.7

36. ПРИЛОЖЕНИЕ ТЕОРИИ КОДИРОВАНИЯ И КРИПТОГРАФИЯ

Нго В. Ш. студент гр.373904, Русина Н.В., аспирант

Белорусский государственный университет информатики и радиоэлектроники г. Минск, Республика Беларусь

Ефремов А.А. – канд. экон. наук, доцент каф. ЭИ

Аннотация. В данной работе рассматривается роль криптографии, шифрования и хеш-функций в информационной безопасности и их применение. Криптография является неотъемлемой частью защиты данных, обеспечивая конфиденциальность, целостность, аутентификацию и невозможность отказа от авторства.

Ключевые слова: Шифрование, Дешифрование, Криптографический ключ, Симметричное шифрование, Ассиметричное шифрование, Хэш-функция, Цифровая подпись, Протоколы безопасности, Блокчейн.

Введение. Криптография — это отрасль математики и информационной безопасности, занимающаяся разработкой алгоритмов, протоколов и систем, которые обеспечивают секретность и защиту информации. Она включает в себя шифрование, хэширование, создание цифровых подписей и разработку других методов защиты данных от несанкционированного доступа и использования. Основная цель криптографии — обеспечение конфиденциальности, целостности, аутентификации и невозможности отказа от отправленных сообщений. В современном мире, где цифровая безопасность является критически важной, криптография играет ключевую роль в защите личной и корпоративной информации, а также в обеспечении безопасности электронных транзакций.

Для базового понимания темы криптографии необходимо знать значение ключевых для данной науки терминов, которыми являются:

Шифрование – это процесс преобразования информации в защищённый формат.

Дешифрование – это процесс восстановления зашифрованной информации в исходный, понятный формат.

Криптографический ключ – Это параметр, который определяет конкретное преобразование открытого текста в зашифрованный и наоборот. Ключи бывают разных типов и размеров, и их безопасность зависит от сложности и способа их создания.

Симметричное шифрование – это метод шифрования, где один и тот же ключ используется для шифрования и дешифрования. Примеры симметричных алгоритмов включает AES (Advanced Encryption Standard). [1]

Асимметричное шифрование – это метод шифрования, использующий пару ключей — публичный и приватный. Публичный ключ может быть свободно распространён и используется для шифрования, в то время как приватный ключ держится в секрете и используется для дешифрования. Примеры включают RSA (Rivest–Shamir–Adleman) и ECC (Elliptic Curve Cryptography). [1]

Хэш-функция – это функция, преобразующая входные данные в строку фиксированной длины.

Цифровая подпись – метод подтверждения подлинности и целостности информации. [2]

Протоколы безопасности – это стандарты и системы, обеспечивающие безопасность передачи данных.

Блокчейн – это технология распределённого реестра, использующая криптографические методы для обеспечения безопасности и целостности данных.

Основная часть. Безусловно, одним из самых важных аспектов криптографии является шифрование, которое, к слову, часто используется как синоним первого, что не совсем корректно. С него и будет начат рассказ о криптографии. Двумя самыми популярными способами шифрования являются AES (Advanced Encryption Standard) и RSA (Rivest–Shamir–Adleman).

AES — это современный алгоритм симметричного блочного шифрования, который используется для защиты цифровой информации. Он был выбран правительством США в качестве стандарта после открытого конкурса и заменил предыдущий стандарт DES. Процесс шифрования с использованием AES является сложным и многоэтапным, обеспечивая высокий уровень безопасности. Всё начинается с выбора ключа шифрования, который определяет уникальность каждого процесса шифрования. Этот ключ может быть различной длины: 128, 192 или 256 бит. После выбора ключа он подвергается процедуре расширения, в результате которой генерируется серия раундовых ключей для каждого этапа шифрования. Далее, исходные данные, которые нужно

зашифровать, разбиваются на блоки по 128 бит. Если объем данных не делится нацело на размер блока, последний блок дополняется до необходимого размера, чтобы обеспечить целостность шифрования. [3]

Первый этап шифрования — это добавление начального ключа к блоку данных. Это делается путем применения операции XOR между каждым битом блока данных и битом начального раундового ключа. Это простая, но эффективная операция, которая начинает процесс преобразования данных.

Затем начинается серия раундов шифрования, которая является сердцем AES. Количество раундов зависит от длины ключа: 10 раундов для ключа длиной 128 бит, 12 раундов для 192-битного ключа и 14 раундов для 256-битного ключа. В каждом раунде выполняются четыре основных операции:

Первая операция — SubBytes, где каждый байт блока данных заменяется на соответствующий байт из заранее определенной таблицы подстановок, называемой S-box. Это обеспечивает нелинейность шифрования и делает его более устойчивым к атакам.

Вторая операция — ShiftRows, при которой строки блока данных циклически сдвигаются на определенное количество позиций. Это увеличивает диффузию данных и способствует более равномерному распределению информации по всему блоку.

Третья операция — MixColumns, во время которой столбцы блока данных перемешиваются с использованием специальной математической операции. Это дополнительно увеличивает сложность шифра и делает его более устойчивым к попыткам взлома.

Четвертая операция — AddRoundKey, где к блоку данных снова применяется операция XOR, на этот раз с раундовым ключом текущего раунда. Это завершает раунд шифрования, и блок данных готов к следующему раунду преобразований.

После выполнения всех раундов, в последнем раунде опускается этап MixColumns, и выполняются только SubBytes, ShiftRows и AddRoundKey. В результате получается зашифрованный текст, который теперь можно безопасно хранить или передавать. Для восстановления исходных данных получатель должен выполнить процесс дешифрования, который является обратным шифрованию и требует того же ключа шифрования.

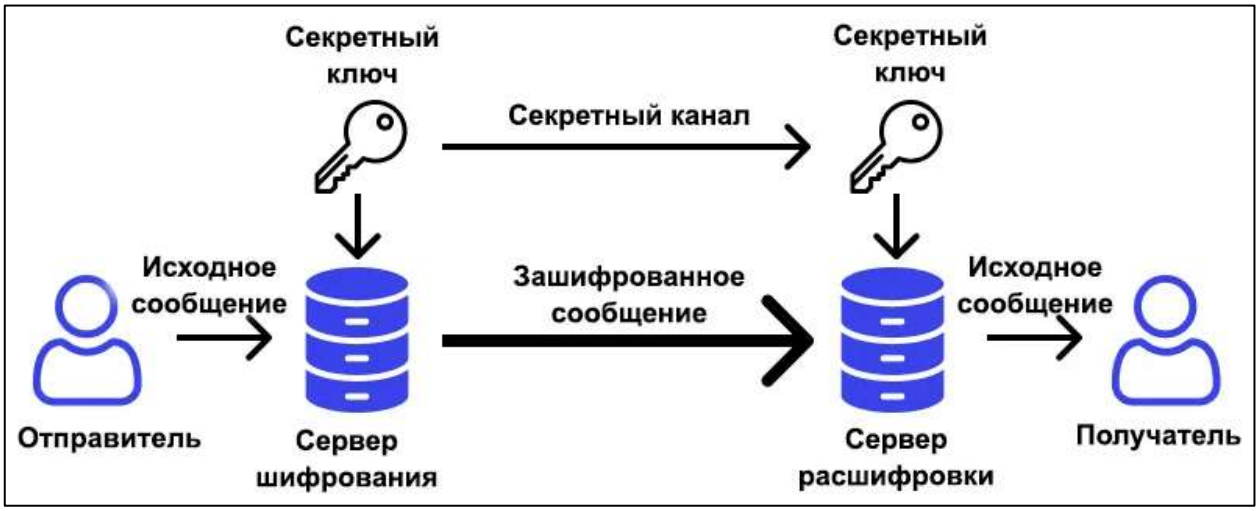

Рисунок 1 – Схема работы AES

RSA (Rivest–Shamir–Adleman) – это асимметричный алгоритм шифрования, который использует пару ключей — публичный и приватный. RSA широко применяется для защиты данных, передаваемых через Интернет, и является стандартом для цифровых подписей. Алгоритм RSA начинается с создания пары ключей: открытого и закрытого. Открытый ключ используется для шифрования сообщений и включает в себя два числа — модуль (n) и открытую экспоненту (e). Закрытый ключ, который держится в секрете, также содержит модуль (n) и закрытую экспоненту (d), которая вычисляется так, чтобы быть мультипликативно обратной к (e) по модулю ((p-1)(q-1)), где (p) и (q) — простые числа, использованные для получения (n). [4]

Когда кто-то хочет зашифровать сообщение, он преобразует текст сообщения в числовое значение, которое затем возводится в степень (e) и берется по модулю (n). Это преобразование

превращает исходное сообщение в зашифрованный текст, который можно безопасно передать. Только обладатель соответствующего закрытого ключа может расшифровать это сообщение.

Для расшифровки зашифрованного текста, получатель использует свой закрытый ключ. Зашифрованный текст возводится в степень закрытой экспоненты (d) и берется по модулю (n), что восстанавливает исходное числовое значение сообщения. Этот процесс обратим только для того, кто знает закрытый ключ, что делает RSA надежным способом шифрования для обеспечения конфиденциальности передачи данных.

Таким образом, RSA обеспечивает безопасный обмен информацией, позволяя отправителю шифровать сообщения открытым ключом, который может быть широко распространен, в то время как только получатель с закрытым ключом может их расшифровать. Это основано на математической сложности факторизации больших чисел, что является краеугольным камнем безопасности RSA. Также в алгоритме RSA для создания цифровой подписи используются хэш-функции, на которых в данной научной работе я хотел бы заострить внимание. [5]

Переходя от алгоритмов шифрования к хэш-функциям, стоит отметить, что они являются ещё одним критически важным элементом криптографии. Хотя шифрование, как в случае с AES и RSA, фокусируется на преобразовании информации в формат, который можно расшифровать только с соответствующим ключом, хэш-функции служат другой цели. Они создают уникальный “отпечаток” данных, который позволяет проверить их целостность без возможности восстановления исходной информации. Это особенно полезно для проверки подлинности данных и создания цифровых подписей, где важно убедиться, что данные не были изменены после их отправки. Таким образом, хэш-функции дополняют шифрование, обеспечивая дополнительный уровень безопасности в цифровом мире.

Хэш-функция — это алгоритм, который принимает входные данные произвольной длины и преобразует их в выходные данные фиксированной длины, называемые хэшем. Этот процесс необратим, то есть по хэшу нельзя восстановить исходные данные. Хэш-функции обладают несколькими важными свойствами: [7]

Детерминизм: одинаковые входные данные всегда приводят к одному и тому же хэшу.

Быстрое вычисление: хэш можно быстро вычислить для любого объема входных данных.

Стойкость к коллизиям: очень маловероятно, что два разных набора входных данных приведут к одному и тому же хэшу.

Лавинный эффект: небольшие изменения во входных данных приводят к значительным и непредсказуемым изменениям в хэше.

Примеры использования хэш-функций:

1. Хранение паролей: Вместо сохранения паролей в открытом виде, системы обычно хранят хэши паролей. Когда пользователь вводит пароль, он хешируется и сравнивается с хэшем, хранящимся в базе данных.

2. Цифровые подписи: Хэш-функции используются для создания уникального отпечатка документа, который затем шифруется с использованием закрытого ключа для создания цифровой подписи.

3. Блокчейн: В блокчейне хэши используются для создания уникального идентификатора каждого блока, что обеспечивает целостность и безопасность цепочки блоков

В современном мире одной из наиболее распространенных хэш-функций является SHA-256, которая стоит в ряду стандартов безопасности и используется во многих криптографических протоколах и системах. SHA-256 принадлежит к семейству хэш-функций SHA-2 и обеспечивает высокую стойкость к коллизиям и лавинный эффект, что делает её надежным инструментом для защиты данных.

Принцип работы SHA-256: SHA-256 принимает входные данные и выполняет серию операций, преобразуя их в уникальный хэш фиксированной длины 256 бит. Процесс начинается с предварительной обработки данных, включая добавление битов до достижения определенной длины и добавление длины исходного сообщения в конец. Затем данные разбиваются на блоки, каждый из которых обрабатывается через ряд раундов, включающих битовые операции и логические функции. В результате получается хэш, который является уникальным для каждого набора входных данных. [6]

SHA-256 широко используется в блокчейне, например, в Bitcoin. Каждая транзакция в блокчейне хешируется с использованием SHA-256, и этот хэш используется для создания уникального идентификатора блока. Это обеспечивает неизменность и целостность всей цепочки блоков, так как любое изменение в одном блоке потребует пересчета хэшей всех последующих блоков.

В современном мире, помимо SHA-256, используются различные хэш-функции, каждая из которых имеет свои особенности и области применения:

MD5: Используется для проверки целостности файлов при загрузке программного обеспечения. Применяется в некоторых базах данных для хранения хешей паролей.

SHA-1: Находит применение в некоторых старых системах и приложениях, например, в прошлом использовался для аутентификации на веб-сайтах.

bcrypt: Применяется для защищенного хранения паролей, например, в сервисе end-to-end шифрованной почты Tutanota. Используется в различных языках программирования и фреймворках, таких как Node.js, для хеширования паролей.

SHA-512: Применяется в .NET для вычисления хэша данных.

Используется в операционных системах на основе Linux для генерации хэшей файлов.

Эти примеры демонстрируют, что, несмотря на наличие более новых и безопасных алгоритмов, старые хэш-функции все еще находят свое применение в определенных областях. Однако важно отметить, что с течением времени и развитием технологий предпочтение отдается более современным и безопасным методам хеширования.

Вывод: Криптография и шифрование играют ключевую роль в обеспечении безопасности в цифровом мире, а хэш-функции, такие как SHA-256, усиливают эту безопасность, предоставляя надежные средства для проверки целостности и подлинности данных. Они позволяют нам защищать информацию от несанкционированного доступа и поддерживать доверие в электронных транзакциях и коммуникациях. Использование хэш-функций в таких технологиях, как блокчейн, подчеркивает их значимость и вклад в развитие современных систем безопасности.

Список использованных источников:

1. GeekBrains [Электронный ресурс] Режим доступа: https://gb.ru/blog/algoritmy-shifrovaniya/ Дата доступа: 08.04.2024
2. Myfin.by [Электронный ресурс] Режим доступа: https://myfin.by/wiki/term/elektronnaya-cifrovaya-podpis Дата доступа: 08.14.2024
3. GeeksforGeeks [Электронный ресурс] Режим доступа: https://www.geeksforgeeks.org/advanced-encryption-standard-aes/ Дата доступа: 08.04.2024
4. GeeksforGeeks [Электронный ресурс] Режим доступа: https://www.geeksforgeeks.org/rsa-algorithm-cryptography/ Дата доступа: 08.04.2024
5. Habr.com [Электронный ресурс] Режим доступа: https://habr.com/ru/articles/729260/ Дата доступа: 08.04.2024
6. Habr.com [Электронный ресурс] Режим доступа: https://habr.com/ru/articles/534014/ Дата доступа: 08.04.2024
7. SkillFactory [Электронный ресурс] Режим доступа: https://blog.skillfactory.ru/glossary/heshirovanie/ Дата доступа: 08.04.2024