Федеральное государственное автономное образовательное учреждение высшего образования

«Российский университет дружбы народов имени Патриса Лумумбы»

Факультет физико-математических и естественных наук

Кафедра теории вероятности и кибербезопасности

РЕФЕРАТ
на тему «СИММЕТРИЧНЫЕ КРИПТОСИСТЕМЫ. ОБЗОР, ВИДЫ, ПРИМИНЕНИЕ»

дисциплина: Основы информационной безопасности

Студент: Ощепков Д. В.

Группа: НКАбд-02-22

№ ст. билета: 1132226442

МОСКВА

2024 г.

Оглавление

1. # Список сокращений

DES - Data Encryption Standard (Стандарт шифрования данных) - это симметричный блочный шифр, который был разработан в 1970-х годах и использовался для шифрования данных. DES был одним из наиболее широко используемых алгоритмов шифрования, хотя его использование сейчас считается устаревшим из-за относительно короткой длины ключа и выявленных уязвимостей.

2. **AES** - Advanced Encryption Standard (Стандарт расширенного шифрования) - это симметричный блочный шифр, который был разработан в результате конкурса, организованного Национальным институтом стандартов и технологий (NIST) в 2001 году. AES является наиболее распространенным алгоритмом шифрования на сегодняшний день и предоставляет высокий уровень безопасности благодаря использованию более длинных ключей и более сложных алгоритмов.
3. **RC4** - Rivest Cipher 4 (Шифр Ривеста 4) - это алгоритм поточного шифрования, который был разработан Рональдом Ривестом в 1987 году. RC4 широко использовался в различных протоколах безопасности, таких как SSL и WEP, но из-за выявленных уязвимостей в его безопасности его использование сейчас не рекомендуется.
4. **ECC** - Elliptic Curve Cryptography (Криптография на эллиптических кривых) - это метод асимметричного шифрования, который использует математические свойства эллиптических кривых для создания криптографических ключей. ECC обладает высокой стойкостью к атакам и требует меньших вычислительных ресурсов по сравнению с традиционными алгоритмами, такими как RSA.
5. **XOR** - Exclusive OR (Исключающее ИЛИ) - это логическая операция, которая возвращает истинное значение только в том случае, если оба входных бита различны. В криптографии XOR часто используется для комбинирования данных с ключом шифрования, так как он обладает свойствами инверсии и обратимости, что делает его полезным для шифрования и расшифрования данных.

Введение

Симметричные криптосистемы играют важную роль в обеспечении безопасности информации в современном мире. Они представляют собой класс криптографических алгоритмов, где один и тот же ключ используется для как шифрования, так и расшифрования сообщений. В данном докладе мы рассмотрим основные принципы работы симметричных криптосистем, выделим их виды и рассмотрим практические примеры их применения в различных сферах, от защиты данных до обеспечения конфиденциальности коммуникаций.

Понимание симметричных криптосистем и их возможностей имеет решающее значение для создания эффективных систем безопасности, способных защитить конфиденциальность и целостность информации от несанкционированного доступа. Данный обзор позволит ознакомиться с основными принципами и применением симметричных криптосистем в современном информационном мире.

Два вида криптосистем

Симметричная криптосистема и несимметричная криптосистема представляют собой два основных подхода к шифрованию данных, и их основное отличие заключается в использовании ключей.

1. **Симметричная криптосистема**:
    - В симметричной криптосистеме используется один и тот же секретный ключ для шифрования и расшифрования данных.
    - Ключ является секретным и должен быть известен как отправителю, так и получателю сообщения.
    - Примеры симметричных алгоритмов включают DES, AES, идея шифрования, RC4 и т. д.
2. **Несимметричная криптосистема (также известная как асимметричная криптосистема)**:
    - В несимметричной криптосистеме используются два различных ключа: открытый и закрытый.
    - Открытый ключ используется для шифрования данных, в то время как закрытый ключ используется для их расшифрования.
    - Открытый ключ может быть распространен публично, в то время как закрытый ключ остается секретным и известным только владельцу.
    - Примеры несимметричных криптосистем включают RSA, ECC, DSA и т. д.

Основные преимущества и недостатки каждого типа криптосистем:
Симметричная криптосистема:

- **Преимущества**:
- Более высокая скорость работы шифрования и расшифрования.
- Простота реализации и использования.
- **Недостатки**:
- Требуется безопасный канал для обмена секретным ключом.
- Проблемы с распределением ключей при обмене данными с большим количеством сторон.

Несимметричная криптосистема:

- **Преимущества**:
- Отсутствие необходимости в безопасном обмене секретными ключами.
- Поддержка цифровой подписи и аутентификации.
- **Недостатки**:
- Более низкая скорость работы из-за сложности математических операций.
- Требуется более высокая вычислительная мощность для работы с большими объемами данных.

Выбор между симметричной и несимметричной криптосистемой зависит от требований конкретного применения, уровня безопасности, доступных ресурсов и других факторов.

Концептуальное устройство симметричных криптосистем

1. **Генерация ключа**: перед началом обмена данными стороны должны сгенерировать секретный ключ, который будет использоваться для шифрования и расшифрования сообщений.
2. **Канал передачи ключа**: секретный ключ должен быть передан от отправителя к получателю безопасным способом. Обычно для этого используется защищенный канал связи, например, защищенное подключение к Интернету (SSL/TLS) или физическая передача ключа через надежного курьера.
3. **Обмен ключами**: для обмена ключами могут использоваться различные протоколы, такие как протокол Диффи-Хеллмана, протокол аутентификации Шнорра и другие. Эти протоколы позволяют сторонам безопасно договориться о секретном ключе, не передавая его напрямую по открытому каналу связи.
4. **Защита ключа**: после передачи ключа его необходимо защитить от несанкционированного доступа. Это может включать в себя его шифрование с помощью другого секретного ключа или физическую защиту хранилища ключей.
5. **Использование ключа и шифрование сообщений**: получив ключ, стороны могут использовать его для шифрования и расшифрования сообщений. Шифрование осуществляется с использованием выбранного симметричного шифра. Существует множество видов шифров, речь о них пойдет ниже.

Зачем нужны алгоритмы шифрования?

1. **Шифрование информации**: основная задача алгоритмов шифрования - преобразование открытого текста (например, сообщения или файлов) в шифрованный текст с использованием секретного ключа. Это позволяет скрыть содержимое сообщения от несанкционированного доступа.
2. **Расшифровка информации**: после того как зашифрованный текст был передан получателю, алгоритмы шифрования также обеспечивают процесс расшифровки, то есть преобразование шифрованного текста обратно в открытый текст с использованием того же самого ключа, который использовался для шифрования.
3. **Обеспечение конфиденциальности**: целью алгоритмов шифрования является обеспечение конфиденциальности информации, путем предотвращения несанкционированного доступа к содержимому сообщений или данных. Шифрование делает информацию непонятной для третьих лиц, которые не имеют доступа к секретному ключу.
4. **Защита от подделки и изменений**: некоторые алгоритмы шифрования также могут обеспечивать аутентификацию и целостность данных, позволяя получателю проверить подлинность сообщения и убедиться, что оно не было изменено в процессе передачи.
5. **Обеспечение безопасности в сети**: шифрование используется для защиты данных в сетевых коммуникациях, таких как передача электронной почты, банковские транзакции, обмен файлами и другие сетевые операции. Оно помогает предотвратить перехват и прослушивание конфиденциальной информации третьими лицами.

Алгоритм DES

DES (Data Encryption Standard) - это симметричный блочный шифр, разработанный в 1970-х годах и используемый для шифрования данных. Давайте рассмотрим основные шаги работы алгоритма DES:

1. **Исходные данные**: входные данные, которые требуется зашифровать, разбиваются на блоки фиксированной длины (обычно 64 бита). Если длина входных данных не кратна 64 битам, то используется дополнение.
2. **Начальная перестановка (Initial Permutation)**: каждый блок данных проходит через начальную перестановку, в результате которой биты блока переупорядочиваются в соответствии с заданной таблицей перестановки.
3. **Раунды (Rounds)**: DES состоит из 16 раундов (повторяющихся операций), каждый из которых включает в себя несколько этапов:
    - **Расширение (Expansion)**: блок данных расширяется до 48 бит с использованием таблицы расширения.
    - **Смешивание с ключом (Key Mixing)**: расширенный блок данных комбинируется с раундовым ключом с использованием операции XOR.
    - **Преобразование S-бокс (S-Box Transformation)**: результат смешивания подается на вход нелинейной S-бокс функции, которая заменяет каждые 6 бит на 4 бита, используя таблицы замен.
    - **Перестановка P (Permutation)**: полученные 32 бита подвергаются перестановке с использованием таблицы перестановки P.
    - **Обмен половин (Swap)**: половины блока данных меняются местами перед следующим раундом.
4. **Конечная перестановка (Final Permutation)**: после завершения всех раундов блок данных проходит через конечную перестановку, которая обратна начальной перестановке.
5. **Зашифрованные данные**: Полученные блоки данных после окончания конечной перестановки являются зашифрованным текстом.

6. **Расшифровка**: расшифровка данных с использованием DES выполняется таким же образом, как и шифрование, но с использованием обратных ключей для каждого раунда.

Хотя DES был стандартом шифрования на протяжении десятилетий, его использование стало нежелательным из-за относительно низкой длины ключа (56 бит) и выявленных уязвимостей. В настоящее время рекомендуется использовать более современные и криптографически стойкие алгоритмы, такие как AES (Advanced Encryption Standard).

Алгоритм AES

AES (Advanced Encryption Standard) - это симметричный блочный шифр, который является одним из самых распространенных алгоритмов шифрования данных. Он был разработан как стандарт шифрования для защиты конфиденциальной информации и обеспечения безопасности в различных информационных системах. AES использует секретный ключ для шифрования и расшифрования данных, и обеспечивает высокий уровень безопасности благодаря своей криптографической стойкости и эффективности. AES поддерживает различные длины ключей (128, 192 и 256 бит), что позволяет выбирать уровень безопасности в зависимости от требований приложения.

Рассмотрим основные шаги работы алгоритма AES (Advanced Encryption Standard):

1. **Исходные данные**: входные данные, которые требуется зашифровать, разбиваются на блоки фиксированной длины. Размер блока в AES составляет 128 бит (16 байт).
2. **Начальная перестановка (AddRoundKey)**: на первом этапе открытый текст комбинируется с раундовым ключом с помощью операции XOR.
3. **Раунды (Rounds)**: AES состоит из нескольких раундов шифрования, в каждом из которых происходят следующие этапы:
    - **SubBytes**: замена каждого байта блока данных на соответствующий байт из S-бокса (подстановочной таблицы).
    - **ShiftRows**: циклический сдвиг строк блока данных влево на разное количество позиций.
    - **MixColumns**: линейное преобразование столбцов блока данных с использованием матрицы умножения.
    - **AddRoundKey**: комбинация блока данных с раундовым ключом с использованием операции XOR.
4. **Финальный раунд**: Последний раунд шифрования не содержит операции MixColumns, только SubBytes, ShiftRows и AddRoundKey.
5. **Финальный вывод**: После завершения всех раундов шифрования полученный шифрованный текст представляет собой зашифрованную версию исходного открытого текста.

6. **Расшифровка**: расшифрование данных с использованием AES выполняется путем применения обратных операций к шифрованию, включая обратное преобразование MixColumns, обратную замену байтов (InvSubBytes), обратный сдвиг строк (InvShiftRows) и обратную операцию XOR с раундовыми ключами.

Важно отметить, что AES обеспечивает высокую степень безопасности и широко используется в современных системах безопасности благодаря своей криптографической стойкости и эффективности.

Алгоритм RC4

RC4 (Rivest Cipher 4) - это алгоритм поточного шифрования, разработанный Рональдом Ривестом в 1987 году. Он широко использовался в различных протоколах безопасности, таких как SSL и WEP (Wired Equivalent Privacy)

Процесс работы RC4 можно описать следующим образом:

1. **Инициализация S-блока**:
   - Создается массив S-блок (S-box) размером 256 байт (от 0 до 255).
   - Этот массив инициализируется значениями от 0 до 255 в порядке возрастания.
2. **Инициализация ключа**:
   - Ключевая последовательность (ключ) длиной от 40 до 2048 бит представляется в виде байтовой последовательности.
   - Этот ключ используется для перемешивания значений в S-блоке в соответствии с алгоритмом ключевого расписания.
3. **Инициализация состояния**:
   - Создается два указателя, i и j, которые инициализируются значениями 0 и 0 соответственно.
   - Значения в S-блоке переставляются в соответствии с ключом (алгоритм ключевого расписания).
4. **Генерация псевдослучайной последовательности**:
   - В цикле для каждого байта открытого текста или для каждого байта зашифрованного текста генерируется псевдослучайный байт.
   - В каждой итерации значения в S-блоке переставляются, чтобы создать псевдослучайную последовательность.
   - Это происходит путем обмена значений в S-блоке и последующего выполнения операции XOR над определенными байтами.
5. **Шифрование / Расшифрование**:

- Полученная псевдослучайная последовательность комбинируется с открытым текстом или с зашифрованным текстом с использованием операции XOR.
- Это позволяет получить зашифрованный текст или расшифровать зашифрованный текст, соответственно.

6. **Итерации**:
    - Процесс генерации псевдослучайной последовательности и шифрования / расшифрования может быть продолжен многократно для каждого блока данных или для всего сообщения.

Таким образом, RC4 использует ключ и начальное состояние S-блока для генерации псевдослучайной последовательности, которая затем комбинируется с данными для шифрования или расшифрования.

Отличия DES и AES, RC4

1. **Тип шифрования**:
    - DES (Data Encryption Standard): DES является блочным шифром с длиной блока 64 бита. Он работает по принципу замены и перестановки битов.
    - AES (Advanced Encryption Standard): AES также является блочным шифром, но с более длинными блоками - 128 бит. Он также работает на основе замены и перестановки битов, но использует более сложные алгоритмы и большее количество раундов для шифрования данных.
    - RC4 (Rivest Cipher 4): RC4 является алгоритмом поточного шифрования, что означает, что он шифрует данные байт за байтом на основе псевдослучайной последовательности. Длина ключа может варьироваться от 40 до 2048 бит.
2. **Длина ключа**:
    - DES: Использует фиксированный ключ длиной 56 бит.
    - AES: Поддерживает ключи длиной 128, 192 и 256 бит.
    - RC4: Поддерживает переменную длину ключа от 40 до 2048 бит.
3. **Скорость работы**:
    - DES: DES характеризуется сравнительно низкой скоростью работы на современных устройствах из-за своей сложной структуры и небольшой длины блока.
    - AES: AES обычно работает быстрее DES благодаря более эффективным алгоритмам и большему размеру блока.
    - RC4: RC4 обычно работает быстрее DES и AES из-за своего поточного характера и относительной простоты алгоритма.
4. **Безопасность**:
    - DES: DES считается устаревшим и уязвимым к современным криптоаналитическим атакам из-за короткой длины ключа.

- AES: AES считается более безопасным по сравнению с DES благодаря использованию более длинных ключей и более сложных алгоритмов.
- RC4: RC4 имеет некоторые известные уязвимости и не рекомендуется для использования в новых приложениях из-за этих проблем.

5. **Использование**:
   - DES: DES широко использовался в прошлом, но сейчас его использование устарело из-за низкой безопасности.
   - AES: AES является одним из самых популярных алгоритмов шифрования и широко используется в различных приложениях и протоколах безопасности.
   - RC4: RC4 ранее использовался в протоколах безопасности, но его использование сейчас также считается устаревшим из-за известных уязвимостей.

Методы Атак

1. **Подбор ключа (Brute Force)**:
    - Этот метод атаки предполагает перебор всех возможных комбинаций ключей для расшифровки зашифрованных данных.
    - Для DES с его относительно коротким ключом в 56 бит, атака полным перебором ключа является практически выполнимой задачей при использовании современных вычислительных ресурсов.
    - Для AES с более длинными ключами (128, 192 или 256 бит), метод полного перебора ключа становится намного более сложным и требует огромных вычислительных ресурсов и времени.
2. **Атака по известному открытому тексту (Known Plaintext Attack)**:
    - В этом типе атаки злоумышленник имеет доступ к зашифрованным данным и соответствующим открытым текстам.
    - Злоумышленник использует эту информацию для анализа и атаки на шифр, пытаясь выявить закономерности или слабости в процессе шифрования.
3. **Атака по выбранному открытому тексту (Chosen Plaintext Attack)**:
    - В этом типе атаки злоумышленник может выбирать определенные открытые тексты и наблюдать за соответствующими зашифрованными данными.
    - После этого злоумышленник пытается анализировать полученные данные и искать уязвимости в шифре.
4. **Дифференциальный криптоанализ (Differential Cryptanalysis)**:
    - Этот метод атаки направлен на нахождение характерных различий между зашифрованными текстами, полученными при разных входных данных.
    - Злоумышленник анализирует различия в выходных данных для выявления корреляций и слабостей в алгоритме шифрования.
5. **Линейный криптоанализ (Linear Cryptanalysis)**:
    - Этот метод атаки основан на поиске линейных аппроксимаций между входными и выходными данными алгоритма шифрования.
    - Злоумышленник использует статистические методы для поиска линейных зависимостей, которые могут помочь в раскрытии ключа.

Заключение

В заключении, симметричные криптосистемы играют важную роль в обеспечении безопасности информации путем шифрования и защиты конфиденциальности данных. Они предоставляют эффективные и быстрые методы шифрования, позволяя защищать информацию на различных уровнях, включая хранение, передачу и обработку данных.

Однако существуют некоторые ограничения и проблемы, с которыми сталкиваются симметричные криптосистемы. Одним из главных ограничений является необходимость безопасного обмена секретным ключом между отправителем и получателем перед использованием шифрования. Это может быть сложной задачей, особенно при обмене данными через открытые сети.

Кроме того, симметричные криптосистемы подвержены различным видам криптоаналитических атак, таким как атаки перебором ключа, известного открытого текста и другие. Это подчеркивает важность выбора сильных ключей и алгоритмов шифрования для обеспечения надежной защиты данных.

Несмотря на эти ограничения, симметричные криптосистемы остаются широко используемым и эффективным инструментом для обеспечения конфиденциальности и целостности информации. При правильном выборе алгоритмов и правильном управлении ключами они могут обеспечить высокий уровень защиты данных в различных сферах, включая коммерческие, правительственные и личные приложения.

Список литературы

1. Блог “IT-спец. Денис Курец”
2. "Применение криптографии. Протоколы, алгоритмы и исходный код на языке C" (Applied Cryptography: Protocols, Algorithms, and Source Code in C) Брюса Шнайера
3. "Введение в современную криптографию" (Introduction to Modern Cryptography) Джонатана Каца и Йехуды Линделла