1. В современном мире, где информация играет ключевую роль во всех сферах жизни, шифрование становится неотъемлемой частью обеспечения безопасности и конфиденциальности данных. Важность шифрования проявляется в нескольких ключевых аспектах:

* Защита конфиденциальности данных: В современном цифровом мире огромное количество информации хранится и передается через интернет. Это могут быть личные сообщения, банковские данные, медицинская информация и многое другое. Шифрование помогает защитить эту информацию от несанкционированного доступа и обеспечить конфиденциальность и неприкосновенность данных.
* Безопасность онлайн-транзакций: С увеличением числа онлайн-транзакций, таких как покупки в интернете, банковские переводы и использование электронных кошельков, важно обеспечить безопасность финансовых операций. Шифрование играет ключевую роль в защите финансовых данных от киберпреступников и мошенников.
* Обеспечение безопасности коммуникаций: Как в личной, так и в деловой сфере обмен конфиденциальной информацией часто происходит через электронные коммуникационные каналы, такие как электронная почта, мессенджеры и видеоконференции. Шифрование обеспечивает защиту переписки от прослушивания и перехвата третьими лицами.
* Защита от кибератак: Кибератаки становятся все более распространенными и сложными, и данные становятся одной из основных целей атак. Шифрование помогает защитить данные от таких атак, таких как вредоносное программное обеспечение, перехват сетевого трафика и атаки на центральные серверы.

1. Шифрование — это способ преобразования данных таким образом, чтобы они не могли быть прочитаны кем-либо, кроме авторизованных сторон. Процесс шифрования превращает обычный текст в зашифрованный с использованием криптографического ключа. Криптографический ключ представляет собой значение, известных и согласованных отправителем и получателем.
2. Шифрование может быть классифицировано по различным критериям, включая тип используемых ключей и принцип работы алгоритмов. Одной из основных классификаций является разделение на симметричное и асимметричное шифрование.

* Симметричное шифрование: В этом виде шифрования один и тот же ключ используется как для шифрования, так и для дешифрования данных. Оба участника процесса обмена данными должны иметь доступ к одному и тому же секретному ключу.
* Асимметричное шифрование: Этот тип шифрования использует пару ключей: открытый и закрытый. Открытый ключ используется для шифрования данных, а закрытый - для их дешифрования. Этот подход обеспечивает возможность безопасного обмена данными без предварительного обмена секретными ключами.

Потоковые:

RC4 (РЦ-четыре) Salsa20 HC-256 (Эйч-Си-двести пятьдесят шесть) WAKE

Блочные:

AES DES 3DES BlowfishTwofish

Асимметричные:

RSA DSA Diffie-Hellman (Диффи-Хеллман) ECC (Эллиптическая криптография)

**Блочное шифрование**

Преимущества:

* Безопасность: Блочные алгоритмы обеспечивают высокий уровень безопасности благодаря сложности структуры и возможности использования больших ключей.
* Структурированный процесс: Блочное шифрование разбивает данные на блоки фиксированного размера, что делает процесс шифрования и дешифрования более структурированным и предсказуемым.
* Параллельная обработка: Некоторые блочные алгоритмы позволяют обрабатывать несколько блоков данных одновременно, что увеличивает скорость обработки данных.

Недостатки:

* Размер блока: Фиксированный размер блока может быть неудобным при работе с данными, размер которых не соответствует размеру блока. Это требует использования дополнительных методов, таких как дополнение данных.

**Потоковое шифрования**

Преимущества:

* Эффективность в передаче потока данных: Потоковое шифрование может работать с потоком данных, не требуя их разбиения на блоки, что делает его эффективным для непрерывного потока данных, таких как поток видео или аудио.
* Отсутствие дополнения данных: Поскольку потоковое шифрование работает с непрерывным потоком данных, оно не требует дополнения данных, что может быть необходимо в блочных алгоритмах.

Недостатки:

* Сложность управления ключами: Потоковое шифрование требует генерации потоков ключей, что может быть сложным и требовать дополнительных вычислительных ресурсов.
* Уязвимость к потере синхронизации: Если происходит потеря синхронизации между генератором ключей и шифруемыми данными, это может привести к потере данных или ошибке дешифрования.
* Меньшая безопасность: Потоковые шифры могут быть менее безопасными, чем блочные, из-за возможности предсказания следующего бита в потоке шифрования при использовании слабого алгоритма генерации ключей.
* Выбор между блочным и потоковым шифрованием зависит от конкретного применения и требований к безопасности и эффективности обработки данных.

1. У AES блок 128 бит. Ключ может быть 128 бит (10 раундов), 192 бита (12 раундов) и 256 бит (14 раундов)

SubBytes заменяет каждый элемент матрицы состояния соответвующим элементом таблицы SBox: sij = SBox[sij]. Преобразование SubBytes обратимо. Обратное к нему находится с помощью таблицы InvSBox.

ShiftRows сдвигает i-ую строку матрицы s на i позиций влево, считая i с нуля. Обратное преобразование InvShiftRows сдвигает строки вправо.

MixColumns умножает каждый столбец матрицы s слева на особую матрицу размера 4×4 (Хотя конечные значения могут быть разными, чем исходные, смешивание происходит в том смысле, что каждый новый байт зависит от комбинации байтов исходного столбца. Это способствует диффузии (распространению) изменений по всему блоку данных, что является важной частью криптографической стойкости шифрования. Таким образом, хотя значения меняются, смешивание относится к идее комбинирования информации в столбце, а не к простому перемешиванию значений.)

GF(2^8) или поле Галуа над двоичным расширением - это математическая структура, которая играет важную роль в криптографии, включая алгоритм AES. Она используется для операций над байтами данных, таких как замена байтов (SubBytes) и смешивание столбцов (MixColumns) в AES.

Определение: Поле Галуа GF(2^8) состоит из 256 элементов (байтов) от 0x00 до 0xFF.

Операции в этом поле выполняются над элементами с использованием арифметики по модулю 2 (XOR и AND).

Представление элементов: Каждый элемент поля Галуа можно представить в виде многочлена степени не выше 7 над полем GF(2). Например, байт 0xAE представляется как x^7 + x^5 + x^4 + x^3 + x.

* Сложение и вычитание: Сложение и вычитание элементов в GF(2^8) выполняется побитово через XOR. Например, 0xAE XOR 0x53 равно 0xFD.
* Умножение: Умножение в GF(2^8) более сложно и требует использования таблицы умножения, называемой таблицей Галуа. В этой таблице каждому элементу соответствует строка, представляющая умножение на этот элемент. Например, умножение 0xAE на 0x2 равно 0x47.
* Обратный элемент: Каждый ненулевой элемент в GF(2^8) имеет обратный элемент относительно умножения. Обратный элемент для элемента a - это такой элемент b, что a \* b = 1.

Поле Галуа GF(2^8) используется в AES для обеспечения нелинейности и криптографической стойкости операций над данными. Оно позволяет выполнять операции шифрования и дешифрования, обеспечивая при этом высокую стойкость к различным атакам.

1. LFSR (Linear Feedback Shift Register) - это регистр сдвига с линейной обратной связью, который используется в криптографии и генерации псевдослучайных последовательностей. Он представляет собой регистр, в котором биты сдвигаются на одну позицию влево на каждом такте, а новый бит вводится с помощью линейной комбинации предыдущих битов.

Алгоритм А5/1. Сдвигами управляет специальная схема, организующая на каждом шаге смещение как минимум двух регистров, что приводит к их неравномерному движению. Ключевая последовательность формируется путём операции XOR над выходными битами регистров.

1. Криптосистема, разработанная Рабином, подобно RSA основывается на трудной проблеме факторизации больших целых чисел или на проблеме извлечения квадратного корня по модулю составного числа n = р\*q. Устойчивость к факторизации: Когда р и q примерно равны, произведениеп = р х q будет близко к квадрату среднего из них, что делает факторизацию П на простые множители более сложной задачей.

Главным недостатком криптосистемы Рабина является то, что неизвестно который из четырех результатов, m1, m2, m3, m4 равен исходному mi. Если сообщение написано по-русски, выбрать правильное mi, нетрудно. С другой стороны, если сообщение является потоком случайных битов, способа определить, какое mi, – правильное, нет. Одним из способов решить эту проблему служит добавление к сообщению перед шифрованием известного заголовка.

1. Независимо от способа реализации для современных криптографических систем защиты информации сформулированы некоторые общепринятые требования.

1. Должно выполняться правило Керкгоффа, т.с. криптостойкость шифра должна обеспечиваться не секретностью алгоритма, а секретностью ключа.

2. Алгоритм должен быть вычислительно стойким, т.е. его вскрытие может быть осуществлено только решением задачи полного перебора ключей.

Таким образом, алгоритм должен противостоять всем известным на текущий момент криптоаналитическим атакам.

Практическая стойкость алгоритма может быть обеспечена выполнением следующих требований:

• отсутствие статистической зависимости между входной и выходной последовательностями;

• выходная последовательность по своим статистическим свойствам должна быть

похожа на истинно случайную;

• при неизменном входном сообщении незначительное изменение ключа должно приводить к существенному изменению выходной последовательности (лавинный эффект по ключу);

• при неизменном ключе незначительное изменение входного сообщения должно приводить к существенному изменению выходной последовательности (лавинный эффект по входному тексту);

• отсутствие простой и легко устанавливаемой зависимости между ключами, последовательно используемыми в процессе шифрования.

3. Структура алгоритма должна быть ясной, простой и обоснованной, что, во-первых, облегчало бы изучение алгоритма специалистами, а во-вторых, гарантировало бы отсутствие внедренных авторами «закладок», особенностей структуры алгоритма, которыми его авторы могли бы воспользоваться в злонамеренных целях.

4. У алгоритма должны отсутствовать слабые и эквивалентные ключи.

5. Алгоритм должен позволять эффективные реализации, в том числе и аппаратные:

• скорость шифрования должна быть высокой на всех потенциальных аппаратных платформах (от 8- до 64-битных);

• структура алгоритма должна позволять распараллеливание операций в многопроцессорных системах и аппаратных реализациях;

• алгоритм должен предъявлять минимальные требования к оперативной и энергозависимой памяти.

6. Стоимость шифрования должна быть согласована со стоимостью защищаемой информации.