1. На примере своего варианта реализации практического задания пояснить свойства симметричности и обратимости сети Фейстеля.

Свойство симметричности заключается в том, что для шифрования и дешифровки применяется один и тот же алгоритм, но с разной последовательностью ключей, а именно, в обратном порядке.

Свойство обратимости заключается в том, что исходные данные можно восстановить с использованием той же функции F и ключами в обратной последовательности. Не нужно использовать обратную ей функцию.

1. Каким способом достигаются эффекты рассеивания и перемешивания?

Эффект рассеивания подразумевает, что изменения в одном бите открытого текста приводят к изменениям в нескольких битах зашифрованного текста. Достигается это применением функции f(l, k), скрывающей статистические свойства исходных данных.

Эффект перемешивания достигается обменами половинок блоков.

Векторы инициализации (IV) играют ключевую роль в режимах шифрования CBC (Cipher Block Chaining) и CFB (Cipher Feedback). Они обеспечивают уникальность и безопасность зашифрованных данных. Рассмотрим основные требования к генерации и хранению IV для этих режимов.

Общие требования к вектору инициализации

1. Уникальность:

В режиме CFB вектор инициализации должен быть уникальным для каждого сообщения. Если IV не уникален, это может привести к раскрытию открытого текста крипто аналитиками.

В режиме CBC уникальность IV не является строгим требованием, но рекомендуется, чтобы избежать потенциальных уязвимостей.

2. Непредсказуемость:

IV должен быть непредсказуемым. Это достигается путем использования случайных или псевдослучайных чисел, что критично для обеспечения семантической безопасности. Повторное использование одного и того же IV с тем же ключом может привести к компрометации данных36.

3. Размер:

Длина IV должна соответствовать размеру блока шифрования (например, 128 бит для AES). Это обеспечивает корректную работу алгоритма шифрования3.

Специфические требования для режимов CBC и CFB

Режим CBC

Генерация: Для первого блока шифруемого сообщения используется IV, который передается вместе с шифротекстом. Это позволяет расшифровщику восстановить данные, используя тот же IV37.

Хранение: IV не нужно хранить в секрете, но его значение должно быть случайным и непредсказуемым, чтобы избежать атак на шифрование4.

Режим CFB

Генерация: Как и в режиме CBC, вектор инициализации должен генерироваться случайным образом. Он должен изменяться для каждого нового сообщения, что может быть реализовано с помощью счетчиков или других методов генерации случайных чисел12.

Хранение: IV может быть передан вместе с первым блоком шифротекста, так как его секретность не требуется. Однако важно следить за тем, чтобы он оставался уникальным на протяжении всего времени жизни ключа24.

Каждый из режимов блочного шифрования (ECB, CBC, CFB, OFB, CTR) имеет свои особенности и области применения, где они могут быть более или менее целесообразными. Ниже представлены рекомендации по использованию каждого режима.

ECB (Electronic Codebook)

Целесообразно:

Простые приложения, где требуется быстрая обработка небольших объемов данных.

Сценарии, где данные не имеют повторяющихся блоков и не требуют высокой безопасности.

Нецелесообразно:

Хранение конфиденциальной информации, так как одинаковые блоки открытого текста создают одинаковые блоки шифротекста, что может привести к утечкам информации.

Приложения, требующие защиты от криптоанализа.

CBC (Cipher Block Chaining)

Целесообразно:

Шифрование файлов и больших объемов данных, где важна безопасность и целостность.

Приложения, требующие защиты от повторного использования данных (например, в банковских системах).

Нецелесообразно:

Высокоскоростные приложения, требующие параллельной обработки данных, так как каждый блок зависит от предыдущего.

CFB (Cipher Feedback)

Целесообразно:

Потоковое шифрование данных, где необходимо шифрование и расшифровка по частям (например, в сетевых протоколах).

Приложения с ограниченной памятью или ресурсами.

Нецелесообразно:

Сценарии с большими объемами данных, где более эффективны другие режимы (например, CTR).

OFB (Output Feedback)

Целесообразно:

Приложения, требующие высокой скорости обработки и возможности параллельного шифрования.

Системы с высоким уровнем ошибок передачи данных, так как ошибки не влияют на последующие блоки.

Нецелесообразно:

Сценарии, где требуется высокая степень безопасности, так как IV может быть уязвим для атак.

CTR (Counter)

Целесообразно:

Высокопроизводительные приложения, требующие параллельной обработки (например, в облачных сервисах).

Сетевые протоколы и системы с большой пропускной способностью.

Нецелесообразно:

Системы с низким уровнем ресурсов или встраиваемые системы без возможности управления счетчиком и IV.

Каждый режим шифрования имеет свои преимущества и недостатки. Выбор подходящего режима зависит от конкретных требований безопасности и производительности в приложении.

Какие требования предъявляются к криптографически стойким генераторам псевдослучайных последовательностей

* Стойкость к предсказанию: Невозможность предугадать следующие значения по предыдущим.
* Непредсказуемость "назад": Невозможность восстановить предыдущие значения, зная текущее состояние.
* Детерминированность: Одно и то же начальное состояние дает одну и ту же последовательность.
* Высокая энтропия начального состояния: Начальное значение должно быть максимально случайным.
* Равномерное распределение: Выходные значения должны быть статистически неразличимы от истинно случайных.
* Устойчивость к атакам с использованием ресурсов: Сложность взлома даже при наличии мощных вычислительных средств.
* Скорость работы: Достаточная производительность для практического применения.
* Сопротивление повторному использованию: Исключение повторного использования одного и того же состояния или seed.
* Проверяемость: Основа на хорошо изученных криптографических примитивах для теоретического обоснования безопасности.

Как определить длину периода псевдослучайной числовой последовательности?

*Из методички*

На чем основывается надежность алгоритма RSA?

Надежность RSA основана на том, что разложение большого составного числа на простые множители является вычислительно сложной задачей, и пока эта проблема остается неразрешимой за разумное время, RSA считается безопасным. Однако важно использовать достаточную длину ключа (≥2048 бит) и следить за развитием новых технологий, таких как квантовые компьютеры.

Краткий ответ: Надежность RSA основана на сложности факторизации больших целых чисел.

Какие преобразования лежат в основе криптосистем с открытым ключом?

Основными преобразованиями в криптосистемах с открытым ключом являются односторонние функции с ловушкой, возведение в степень по модулю, операции с дискретными логарифмами и вычисления на эллиптических кривых

Таблица Виженера и полиалфавитные шифры

Таблица Виженера (также называемая квадратом Виженера или таблицей Тритемия ) представляет собой матрицу размером 26×26 (для латинского алфавита), где каждая строка является циклически смещенной версией предыдущей строки.

Полиалфавитные шифры — это методы шифрования, при которых для каждой буквы открытого текста используется разный алфавит. Этот подход делает шифр более устойчивым к частотному анализу, чем моноалфавитные шифры (например, шифр Цезаря).

Пример полиалфавитного шифра: Шифр Виженера

Шифр Виженера — один из самых известных полиалфавитных шифров. Он использует таблицу Виженера для замены букв открытого текста.

Дифференциальный криптоанализ

Как работает дифференциальный криптоанализ?

Выбор дифференциальной характеристики :

Атакующий выбирает конкретную дифференциальную характеристику, которая наиболее вероятно будет наблюдаться при шифровании.

Генерация дифференциальных пар :

Создаются пары открытых текстов, такие что их XOR равен выбранному входному различию ΔP.

Шифрование пар :

Каждая пара открытых текстов шифруется с использованием неизвестного ключа, получая зашифрованные тексты.

Анализ выходных различий :

Вычисляется XOR зашифрованных текстов и проверяется, соответствует ли оно предсказанному выходному различию.

Определение ключа :

Если наблюдаемые различия совпадают с предсказанными, это позволяет узкому кругу возможных значений ключа. Повторяя процесс с большим количеством пар, можно сузить множество возможных ключей до единственного правильного.