Формирование стойких ключей шифрования – это критически важный аспект криптографической безопасности.

Ключевые аспекты:

Алгоритмы формирования стойких ключей

- Использование случайных или псевдослучайных чисел: Основа создания стойкого ключа – это генерация случайной или псевдослучайной последовательности. Генераторы случайных чисел должны быть криптографически безопасными (CSPRNG).
- Длина ключа: Для современных алгоритмов шифрования, таких как AES, рекомендуемая длина ключа составляет 128 или 256 бит(иногда 192 бита). Большая длина ключа обычно означает большую стойкость, но также увеличивает вычислительные затраты.
- Энтропия: Ключ должен иметь высокую энтропию, то есть быть непредсказуемым. Это исключает легко угадываемые ключи, такие как простые слова или последовательности.

Поиск и исключение слабых ключей

- Тестирование на слабые ключи: Некоторые алгоритмы, например, DES, имеют известные слабые ключи, которые следует избегать. Современные алгоритмы обычно разрабатываются с целью минимизации наличия слабых ключей.
- Анализ устойчивости: Регулярный криптоанализ используемых алгоритмов помогает выявлять потенциальные уязвимости.

Практическая реализация:

- Хранение ключей: Безопасное хранение ключей – также критический аспект. Это может включать аппаратные средства безопасности, например, HSM (Hardware Security Module).
- Обновление ключей: Регулярное обновление ключей помогает поддерживать их безопасность. Это важно особенно в системах, где возможны утечки информации.
- Протоколы управления ключами: Включают процедуры генерации, распределения, хранения, обновления и уничтожения ключей.

В области формирования стойких ключей шифрования важно понимать различие между абсолютной и вычислительной стойкостью. Абсолютная стойкость достигается, когда криптосистема теоретически и практически не может быть раскрыта, даже при наличии у атакующего бесконечных вычислительных ресурсов. Одним из примеров абсолютно стойкого алгоритма является шифр Вернама (одноразовый блокнот). Для абсолютной стойкости ключ должен генерироваться для каждого сообщения, быть статистически надежным, иметь длину, равную или большую длины сообщения, и исходный текст должен обладать избыточностью (что является критерием оценки правильности расшифровки).

В современных криптографических системах обычно используются практически стойкие или вычислительно стойкие системы. Эти системы могут быть теоретически

вскрыты, но на практике это невозможно из-за огромных вычислительных затрат, необходимых для их взлома.

Для определения слабых ключей в шифровании используются различные методы и алгоритмы, которые включают анализ структурных особенностей алгоритма шифрования и исследование поведения алгоритма при различных ключах. Существует необходимость использования образцов для оценки полученных ключей, и проверки ключей на слабость могут увеличивать время выполнения алгоритмов и объем требуемой памяти. На практике, для предотвращения возможности атак, связанных со временем, ключи часто создаются повторно.

Важные понятия:

1. CSPRNG (Cryptographically Secure Pseudorandom Number Generator)

CSPRNG — это тип генератора псевдослучайных чисел, специально разработанный для использования в криптографических приложениях. Он отличается от обычных генераторов псевдослучайных чисел тем, что предлагает более высокий уровень безопасности, так как его вывод очень трудно предсказать. Это критически важно для таких задач, как генерация ключей шифрования и случайных нонсов (одноразовых чисел).

Основные характеристики CSPRNG:

- Непредсказуемость: Невозможность предсказать следующее число в последовательности, исходя из предыдущих чисел.
- Устойчивость к атакам: Способность противостоять попыткам восстановить внутреннее состояние генератора.

Примеры алгоритмов:

1) *Fortuna: Разработанный известными криптографами Нильсом Фергюсоном и Брюсом Шнайером, Fortuna используется для генерации криптографически стойких псевдослучайных чисел. Он разделен на несколько источников энтропии для повышения безопасности и устойчивости к атакам.*
2) *Yarrow: Еще один алгоритм, созданный Брюсом Шнайером и другими, Yarrow предназначен для эффективной и безопасной генерации псевдослучайных чисел. Он использует смесь различных источников энтропии и механизмов обратной связи для обеспечения криптографической стойкости.*

2. PBKDF2 (Password-Based Key Derivation Function 2)

PBKDF2 — это функция, используемая для преобразования пароля в криптографический ключ. Эта технология часто применяется в системах, где необходимо защитить пароли или другие данные, предназначенные для генерации ключей. PBKDF2 использует механизм, называемый "усилением ключа", который включает многократное применение хеш-функции к паролю.

Основные особенности PBKDF2:

- Медленное выполнение: Умышленное замедление процесса создания ключа для затруднения атак силовым перебором.
- Соль (Salt): Использование случайной соли для каждого пароля, чтобы предотвратить атаки по словарю и использование радужных таблиц.

3. HSM (Hardware Security Module)

HSM — это физическое устройство, предназначенное для безопасного генерирования, хранения и управления криптографическими ключами. HSM обеспечивает высокий уровень безопасности по сравнению с программными решениями, так как криптографические операции и ключи никогда не покидают физически защищенное устройство.

Основные преимущества HSM:

- Физическая безопасность: Устойчивость к физическим и логическим попыткам несанкционированного доступа.
- Эффективность и надежность: Ускорение криптографических операций и обеспечение надежного управления ключами.
- Соответствие стандартам: Соответствие глобальным и отраслевым стандартам безопасности, таким как FIPS 140-2.

Использование этих технологий является ключевым в современной криптографии для обеспечения безопасности и надежности шифрования.