Формирование стойких ключей шифрования

Алгоритмы формирования стойких ключей

Генерация случайных чисел:

Стойкость ключей часто начинается с качественной генерации случайных чисел. Используемые алгоритмы должны обеспечивать высокую энтропию, чтобы предотвратить предсказуемость ключей. Криптографические ГПСЧ (генераторы псевдослучайных чисел) являются неотъемлемой частью этого процесса.

Примером может послужить Fortuna - ГПСЧ, обеспечивающий стойкость ключей. Он использует несколько пулов энтропии, собирая случайные данные от различных источников, таких как движение мыши или нажатия клавиш. Fortuna аккумулирует энтропию, усредняя ее для создания более стойких случайных чисел.

Внутренние ключи Fortuna регулярно обновляются, что предотвращает возможные атаки. Хроники событий фиксируют возраст энтропии, что помогает принимать решения об использовании данных. Алгоритм генерирует ключи, обновляя внутреннее состояние при необходимости. Fortuna эффективно использует разнообразные источники энтропии, что делает его хорошим выбором для приложений, требующих стойкости псевдослучайных чисел.

Ключевая длина:

Стойкость ключа напрямую зависит от его длины. Существует общее правило: чем длиннее ключ, тем сложнее его взломать. Рекомендации по длине ключей могут различаться для разных алгоритмов шифрования, например, в AES, рекомендуемая длина ключа составляет 128 или 256 бит(иногда 192 бита).

Асимметричные алгоритмы:

Асимметричные алгоритмы, такие как RSA, требуют пары ключей: открытый и закрытый. Процесс их генерации также крайне важен. В случае RSA, стойкость ключа зависит от сложности факторизации больших простых чисел.

RSA — это асимметричный криптографический алгоритм, используемый для шифрования данных и создания электронных подписей. Он работает на основе математической сложности задачи факторизации больших чисел.

Практические аспекты поиска слабых ключей

Атаки перебором:

Стандартный метод атаки - перебор ключа. Сложность алгоритма может существенно возрастать с увеличением длины ключа. Например, 128-битный ключ считается достаточно стойким для современных вычислительных мощностей.

Атаки по времени:

Атаки, основанные на времени, могут раскрывать информацию о ключе на основе затраченного времени на его обработку. Такие атаки могут быть использованы для поиска слабых мест в алгоритмах.

Слабые ключи в алгоритмах:

Некоторые алгоритмы могут содержать уязвимости или слабости, которые делают ключи менее стойкими. Регулярные обновления и использование актуальных версий криптографических библиотек помогают минимизировать риски.

Допустим, у нас есть алгоритм шифрования блочного типа, например, DES (Data Encryption Standard). В DES используется 56-битный ключ, что считается небезопасным в современных условиях. С использованием современных вычислительных мощностей, атаки перебором могут быстро проверить все возможные комбинации 2^56 ключей.

Квантовые атаки:

С развитием квантовых вычислений, некоторые современные алгоритмы могут оказаться уязвимыми. Развитие квантовой криптографии, такой как алгоритмы на основе квантовых ключей, может стать ответом на эту угрозу.

Практическая реализация

Хранение ключей:

Безопасное хранение ключей – также критический аспект. Это может включать аппаратные средства безопасности, например, 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.

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.

Абсолютная и вычислительная стойкость

В сфере формирования надежных шифровальных ключей важно учитывать различие между абсолютной и вычислительной стойкостью. Абсолютная стойкость достигается, когда криптосистема теоретически и практически остается неприступной, даже при наличии бесконечных вычислительных ресурсов у потенциального злоумышленника.

Примером абсолютно стойкого алгоритма является шифр Вернама, где ключ генерируется для каждого сообщения, обладает статистической надежностью, имеет длину, равную или большую длине сообщения, и требует избыточности в исходном тексте для оценки правильности расшифровки.

В современных криптографических системах обычно используются практически стойкие или вычислительно стойкие методы. Эти системы могут теоретически быть взломаны, но на практике такие атаки сталкиваются с огромными вычислительными затратами, что делает их взлом невозможным в реальных условиях.