Алгоритм с открытым ключом – тема распространения ключей.

Не нужно встречать и через дупло. Можно передать открыто контрагенту.

Как передавать секретные сообщения в такой же схеме.

Почему симметричные ключи остались жить: ключ большой, сложная математика. В реальных потоковых системах никто не использует алгоритмы с открытым ключом. Есть быстрые алгоритмы. Симметричная схема. Расшифровать через симметричную.

2 стороны.

HTTPS

Симметричный алгоритм

Алгоритм Цезаря

Лавинный эффект – определяется диффузией и конфузией

Диффузия – скрывает статистику исходного текста

Конфузия – устраняет линейную зависимость

Способ взлома – использование модели для определенного ключа.

DES/AES – лавинный эффект должен быть не меньше 2-х для ДЕС по требованию США. Там 16 рауднов. Бит 64. К 10-му раунду ДЕС обеспечивает влияние изменение каждого бита на 44 бита.

AES – изменение одного бита в исходном стейте. Мисколд. 1 раунд – изменение 1 бита в исходных данных влияет на 4 байта. Для вычисление нового столба используется … ???

Перемешивание во втором раунде – в сдвиге. Столбец содержит инфу об измен бите. Будет влиять на все 4 столбца. Перемешивание и рез инфа разъедется на все столбцы.

На 2-ом изменение 1 бита изменит 16 байт.

Гостовский алгоритм – на все 16 байт за 8 рауднов будет распространятся изменение каждого битика.

Главная тема – ХЕШ ФУНКЦИЯ как средства аутентификации сообщений.

Свойства:

1. Применимость к блоку произвольной длины
2. При этом результат вычисления хеша должен быть фиксированным.
3. Необратимость.
4. Если а не равно б. То Хеш функции от а и б не должны быть равны.

Главная угроза – это коллизия или коллизия дней рождения. Вероятность небольшая, но есть. Главное с чем борятся и главное, что используют взломщики в хеш функции.

Как можно реализовать фиксированную длину для данных произвольной длины – разбить на блоки нужно размера и после этого результаты смешиваются между собой.

Для чего используется хеш функция – для того чтобы сокрыть данные.

Аутентификация сохраняется. Пароль не хранится. Хранится его хеш. И сравниваются хешы.

Распространенные алгоритмы: MD versus SHA (Message Digest versus Secure )

RSA1 🡪 MD5 (Message Digest)

2004 MD5CRCK – Энтузиасты искали зависимости, которые приводили к одному и тому же хешу.

2 сертификата, которые позволяли для разных документов подобрать одинаковые хешы.

MD5 – 128 бит блоки. Результат хеш функции – 160 бит. То есть ограничение блока аутентификаторов. Это сейчас катастрофически мало, чтобы подобрать разные хешы. Слишком маленький спектр.

SHAO – in 512b, out 160b. Слабый алгоритм. Через 2 месяца был отозван. Через 5 лет взломан французами.

SHA1 in ‘-‘ out ‘-‘. Гугл нашел коллизии. 2015 публичная атака на него. В 2016 почтовые сервисы отказались от его использование.

MD5/SHA1 – официально не безопасные.

SHA2 – стандарт. Набор алгоритмов, объединенных одной логикой. Вх 512\1024 Вых 256\224 или 512\364.

Ксоры и сдвиги. Получаются новые константы. Проводится много раундов. Около 80. Получается единый блок. Отличия в сборе констант. В зависимости от спектра значений.

D 2007 SHA3 ???