Протоколы обмена ключами

Все протоколы обмена ключами делятся на 3 вида:

- Протоколы, основанные на симметричной криптографии
- Протоколы, основанные на асимметричной криптографии
- Протоколы, использующие центр сертификации (доверенный центр) (Трент)

Симметричная криптография – способ шифрования, когда для шифрования и дешифрования применяется один и тот же ключ.
Ассиметричная криптография использует 2 ключа: публичный и приватный. Публичный передается по незащищенному каналу и используется для проверки ЭП и шифрования сообщения. Приватный ключ используется для расшифровки сообщения и генерации ЭП.
Доверенный центр – третья сторона, чья честность неоспорима. Используется для подтверждения подлинности ключей шифрования с помощью сертификатов ЭП.

Нас не будут интересовать протоколы, взаимодействующие с Трентом. Т.к. это явное нарушение задумки распределенных систем. Рассмотрим же остальные протоколы.

Симметричные протоколы

Главным принципом является условие, что передатчик и приемник заранее знают алгоритм шифрования, а также ключ к сообщению. Классические примеры:

- **Простая перестановка**

Простая перестановка без ключа. Сообщение записывается в таблицу по столбцам. После он считывается по строкам для образования шифртекста. Приемник и передатчик должны договориться о размере таблицы - это и будет ключ.

- **Одиночная перестановка по ключу**

Данный подход отличается от предыдущего тем, что колонки таблицы переставляются по ключевому слову, фразе или набору чисел длиною в строку таблицы.

- **Двойная перестановка**

Двойная шифровка с использование второй таблицы другого размера. Лучше всего будет если длины будут взаимно простыми. Кроме того, в первой таблице можно переставлять столбцы, а во второй строки. Можно заполнять таблицу зигзагом или любым другим способом.

- **Перестановка «Магический квадрат»**

Магическими квадратами называются квадратные таблицы со вписанными в их клетки последовательными натуральными числами от 1, которые дают в сумме по каждому столбцу, каждой строке и каждой диагонали одно и то же число.

Протоколы

Wide-Mouth Frog. Простейший протокол обмена ключами. В протоколе принимает участие доверенный центр.

Описание работы протокола:

- Алиса формирует отметку времени и случайный сессионный ключ.
- Алиса отправляет их Тренту, добавив свое имя и зашифровав.
- Трент, используя общий с Алисой секретный ключ, расшифровывает сообщение и проверяет метку времени и идентификатор Боба.
- Трент формирует новую отметку времени и отправляет Бобу сессионный ключ, имя Боба и новую метку в зашифрованном виде.
- Боб получает сообщение. Расшифровывает его общим с Трентом ключом и проверяет первую(?) отметку времени и идентификатор Алисы.
- Теперь у Алисы и Боба есть общий ключ.

Протокол Нидхема-Шрёдера. Данный протокол имеет историческое значение. Он является основой для многих протоколов распространения ключей, использующий доверенный центр. Данный протокол является примером протокола, не использующего временных меток.

Описание работы протокола:

- Алиса выбирает свое число, Боб выбирает свое число.
- Алиса формирует сообщение, состоящее из своего и Боба идентификаторов, а также выбранного ею числа, и отправляет его Тренту.

- Трент формирует сообщение, состоящее из двух частей. Первая состоит из числа Алисы, идентификатора Боба, а также нового ключа, который хотят получить Алиса и Боб. Вторая часть состоит из этого ключа и идентификатора Алисы, но эта часть зашифрован секретным ключом Трента и Боба. Все сообщение (обе части вместе) шифруются секретным ключом Трента и Алисы. Затем отправляется Алисе.
- Алиса расшифровывает сообщение. Найдя своей число в сообщении, она убеждается, что говорила с Трентом. Вторую часть сообщения она прочитать не способна. Пересылает ее Бобу.
- Боб расшифровывает сообщение. Достает оттуда новый ключ и формирует сообщение для Алисы, в котором сообщает ей свое число, шифрованное новым ключом.
- Алиса получает сообщение и число Боба. Меняет (зачем?) его и отправляет Бобу.
- Алиса и Боб владеют общим ключом.

В блокчейне используются асимметричные протоколы, поэтому обзор симметричных ограничится этими двумя.

Асимметричные протоколы

Принципы асимметричного шифрования:

- Нельзя, зная публичный ключ, вычислить приватный за разумное время. При этом механизм генерации ключей является общеизвестным.
- Сообщение, зашифрованное публичным ключом, можно расшифровать только приватным ключом. Механизм шифрования является общеизвестным.
- Владелец ключей никому не сообщает приватный ключ, но передает открытый ключ или делает его общеизвестным.

Протокол Нидхема-Шрёдера. Протокол Нидхема-Шрёдера на асимметричных ключах был опубликован, как и его симметричный родственник, в 1978 году.

Описание работы протокола:

- У Алисы и Боба есть свои публичные процедуры кодирования. Алиса и Боб хотят взаимно идентифицировать друг друга с помощью трех сообщений и используя публичные ключи.

- Алисы выбирает свою часть ключа и формирует сообщение Бобу, состоящее из части ключа и идентификатора Алисы. Сообщение шифруется публичным ключом Боба и отправляется ему.
- Боб расшифровал сообщение. Боб выбирает свою часть ключа и формирует сообщение, состоящее из двух новых ключей (Алисы и Боба), зашифрованное публичным ключом Алисы.
- Алиса получает сообщение. Забирает оттуда свой новый ключ. Отправляет Бобу сообщение содержащее его часть ключа.
- В итоге, оба уверены, что говорили друг с другом и оба знают весь ключ.

Рассмотрим конкретные примеры протоколов с открытым ключом.

RSA

- алгоритм, основывающийся на вычислительной сложности задачи факторизации больших целых чисел. Самая первая система, пригодная для шифрования и цифровой подписи одновременно.

Шифрование сеансового ключа:

Алгоритм:

- Взять *открытый ключ* (e, n) Алисы
- Создать случайный *сеансовый ключ* m
- Зашифровать сеансовый ключ с использованием открытого ключа Алисы:
  $c = E(m) = m^e \mod n$
- Зашифровать сообщение M_A с помощью сеансового ключа симметричным алгоритмом:
  $C = E_m(M_A)$

Алгоритм:

- Принять зашифрованный сеансовый ключ Боба c
- Взять свой *закрытый ключ* (d, n)
- Применить закрытый ключ для расшифровывания сеансового ключа:
  $m = D(c) = c^d \mod n$
- Расшифровать сообщение C с помощью сеансового ключа симметричным алгоритмом:
  $M_A = D_m(C)$

Цифровая подпись:

Алгоритм:

- Взять открытый текст m
- Создать цифровую подпись s с помощью своего секретного ключа $\{d, n\}$:
  $s = S_A(m) = m^d \mod n$
- Передать пару $\{m, s\}$, состоящую из сообщения и подписи.

Алгоритм:

- Принять пару $\{m, s\}$
- Взять открытый ключ $\{e, n\}$ Алисы
- Вычислить прообраз сообщения из подписи:
  $m' = P_A(s) = s^e \mod n$
- Проверить подлинность подписи (и неизменность сообщения), сравнив m и m'

Подлинность цифровой подписи может каждый, имеющий публичный ключ автора.

В данном примере сообщение m не зашифровано, но ничто не мешает предварительно зашифровать его.

DSA

- алгоритм цифровой подписи, основанный на вычислительной сложности взятия логарифмов в конечных полях. Также, как и в RSA, любой может проверить подлинность подписи, имея публичный ключ автора.

Стоит сказать, что фактически в данном алгоритме подписывается не сообщение, а его хэш.

Цифровая подпись:

1. Выбор случайного числа $k \in (0, q)$
2. Вычисление $r = (g^k \mod p) \mod q$
3. Выбор другого k, если $r = 0$
4. Вычисление $s = k^{-1}(H(m) + x \cdot r) \mod q$
5. Выбор другого k, если $s = 0$
6. Подписью является пара (r, s) общей длины $2N$

Проверка подлинности:

1 Вычисление $w = s^{-1} \mod q$
2 Вычисление $u_1 = H(m) \cdot w \mod q$
3 Вычисление $u_2 = r \cdot w \mod q$
4 Вычисление $v = (g^{u_1} \cdot y^{u_2} \mod p) \mod q$
5 Подпись верна, если $v = r$

Применимость в блокчейне

В блокчейне используются два ключа: приватный и публичный. Оба ключа привязаны к программному кошельку и могут быть импортированы в другой по желанию пользователя.

Открытый ключ является адресом (счетом), на который пересылаются деньги. При помощи него выполняется шифрование – создание транзакции. Но без подписи транзакция будет отменена.

Для создания цифровой подписи используется закрытый ключ. Закрытый ключ лучше всего хранить на устройстве, которое не подключено к интернету.

Источники

1. Протокол распределения ключей – Wikipedia
2. RSA - Wikipedia
3. DSA – Wikipedia
4. Публичные ключи и приватные ключи: что это и как они работают – Binance