ЛЕКЦИЯ 11

ПРОТОКОЛЫ РАСПРЕДЕЛЕНИЯ КЛЮЧЕЙ

Основная задача протоколов распределения ключей — выработка абонентами (Алисой и Бобом) общего (не обязательно секретного) ключа. Вместе с тем и Боб, и Алиса должны быть удостоверены в том, что связь ведется именно с собеседником, а не со злоумышленником или подставным лицом.

Все подобные протоколы опираются на существование специального **доверенного центра (Trent)** и основаны на **асимметричной криптографии**.

В приведенных протоколах Trent выступает в качестве адресной книги: в его базе данных имеются открытые ключи всех абонентов, и, в свою очередь, у всех абонентов имеется открытый ключ Trent.

В протоколах распределения в рамках данной лекции используется шифрование на открытых ключах, что позволяет не только шифровать данные, но и заверять их электронной подписью и эту подпись проверять.

Использование асимметричной криптографии приводит к очевидному недостатку: в таком случае сам факт получения зашифрованного сообщения не несет никакой информации, в отличие от получения сообщения, зашифрованного секретным ключом в симметричной криптографии.

1. Протокол DASS

Distributed Authentication Security Service — распределенная служба безопасности и проверки подлинности.

Ниже и далее E обозначает шифрование (Encryption), а S — подпись (Stamp). Нижний индекс обозначает ключ, при помощи которого зашифровано или подписано сообщение. K_P — случайная сеансовая пара открытого и закрытого ключей для асимметричного шифрования, K — симметричный сеансовый ключ.

1. $A \to \{B\} \to T$
   Алиса запрашивает у Trent открытый ключ Боба.

2. $T \to \{S_T(B, K_B)\} \to A$
   Трент присылает Алисе открытый ключ и идентификатор Боба, подписав все это своим открытым ключом.

3. $A \to \{E_K(T_A), S_A(L, A, K_P), S_{K_P}(E_{K_B}(K))\} \to B$
   Алиса проверяет подпись при помощи открытого ключа Trent, после чего генерирует случайную сеансовую пару ключей K_P, метку времени T_A и срок жизни ключа L. Часть данных шифруется открытым ключом Боба и симметричным сеансовым ключом, часть подписывается открытым ключом Алисы и временным сеансовым ключом.

4. $B \to \{A\} \to T$
   Боб запрашивает у Trent открытый ключ Алисы.

5. $T \to \{S_T(A, K_A)\} \to B$
   Trent высылает Бобу подписанный пакет с ключом Алисы и ее идентификатором. Боб проверяет подпись Алисы, извлекает временный ключ, проверяет подпись пакета с симметричным сеансовым ключом и расшифровывает его своим открытым ключом. Зная симметричный сеансовый ключ, Боб расшифровывает метку времени и проверяет релевантность сообщения.

6. $B \to \{E_K(T_B)\} \to A$
   Боб шифрует метку времени симметричным сеансовым ключом и отправляет Алисе. Алиса убеждается в валидности метки времени; происходит взаимная идентификация сторон.

Особенности DASS:

1. Используются **метки времени (Timestamp)**, обеспечивающие устойчивость протокола к атаке повтором. Метка T_B еще актуальна в течение **времени жизни (Lifespan)** L с момента T_A. При взломе злоумышленником сессионного ключа они будут уже недействительны, и атака повтором провалится. В случае генерации им новой метки его плану помешают уже другие шаги протокола.

2. Алиса генерирует все требуемые ключи. Если она не обладает достаточно мощным компьютером, то полученные ключи окажутся ненадежными.

2. Протокол Деннинга – Сакко

1. $A \to \{A, B\} \to T$
   Алиса отправляет Trent идентификаторы.

2. $T \to \{S_T(A, K_A), S_T(B, K_B)\} \to A$
   Trent отсылает Алисе подписанные им пары «идентификатор-открытый ключ».

3. $A \to \{E_{K_B}(S_A(K, T_A)), S_T(A, K_A), S_T(B, K_B)\} \to B$
   Алиса отправляет Бобу зашифрованные его открытым ключом сеансовый ключ и метку времени а так же прикрепляет сюда сообщение Trent. Боб своим закрытым

ключом расшифровывает сеансовый ключ, проверяет метку времени и подписи Trent.

Особенности протокола Деннинга – Сакко:

1. Здесь Алиса вновь генерирует необходимые ключи.
2. Отсутствие в $E_{K_B}(S_A(K, T_A))$ метки получателя (Боба) позволяет ему после получения этого сообщения выдать себя за Алису.

3. Протокол Ву – Лама

1. $A \to \{A, B\} \to T$
   Алиса отправляет Trent идентификаторы.
2. $T \to \{S_T(K_B)\} \to A$
   Trent подписывает открытый ключ Боба и отсылает Алисе.
3. $A \to \{E_{K_B}(A, R_A)\} \to B$
   Алиса проверяет подпись Trent и отправляет Бобу пару «идентификатор-случайное число», зашифрованную открытым ключом Боба.
4. $B \to \{A, B, E_{K_T}(R_A)\} \to T$
   Боб отправляет Trent идентификаторы и случайное число Алисы, зашифрованное открытым ключом Trent.
5. $T \to \{S_T(K_A), E_{K_B}(S_T(R_A, K, A, B))\} \to B$
   Trent отправляет Бобу подписанный открытый ключ Алисы и набор «случайное число-сеансовый ключ-идентификаторы», подписанный Trent и затем зашифрованный открытым ключом Боба.
6. $B \to \{E_{K_A}(S_T(R_A, K, A, B), R_B)\} \to A$
   Боб отправляет Алисе подписанный набор, полученный от Trent, и дополнительное случайное число R_B, предварительно шифруя это все ее открытым ключом.
7. $A \to \{E_K(R_B)\} \to B$
   Алиса расшифровывает данные своим закрытым ключом, проверяет подпись и отправляет Бобу его случайное число, зашифрованное сеансовым ключом. Приняв и расшифровав этот пакет, Боб сможет удостовериться в том, что Алиса знает сеансовый ключ, сгенерированный Trent.

Особенности протокола Ву – Лама:

1. При сверке Алисой отправленное и полученное ею случайные числа должны совпасть.
2. Здесь Trent в дополнение к привычной функции адресной книги выполняет функцию генератора надежных ключей.

3. Данный протокол уязвим к **DDOS-атакам отражением**. Поскольку в качестве R и A можно взять произвольные числа, открывается возможность отправить Бобу множество запросов, которые он перенаправит к Trent. Подписывая, расшифровывая и зашифровывая собщения, доверенный центр выполняет емкие вычисления и может легко быть выведен из строя пришедшей лавиной запросов.

4. Из-за того, что в протоколе Ву – Лама данные пересылаются во всех возможных направлениях между A, B и T, возникшая в одном из каналов связи неполадка вызовет большие затруднения и будет сложна для обнаружения. По этой причине в хорошем протоколе разумно реализовать связи $A \rightleftharpoons T$ и $A \rightleftharpoons B$, оставив при этом обязанность генерации ключей Trent.

4. Эзотерические протоколы

В число **эзотерических протоколов** входят:

1. Доказательство с нулевым разглашением;
2. Слепая подпись;
3. Подбрасывание монетки по телефону.

5. Доказательство с нулевым разглашением

Предположим, Алисе необходимо показать Бобу, что она знает некоторый секрет. При этом Боб должен уметь определять обман со стороны Алисы и не узнать сам секрет. Эти три идеи можно коротко сформулировать так:

1. Полнота;
2. Корректность;
3. Нулевое разглашение.

Проиллюстрировать работу доказательства с нулевым разглашением можно на следующем наглядном примере:

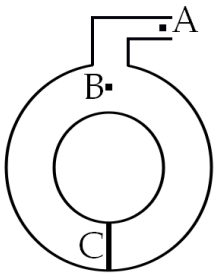

Рис. 11.1

Алиса и Боб стоят у входа в пещеру, внутри которой находится дверь C, запертая на кодовый замок. Боб хочет показать Алисе, что знает код от двери, не раскрывая его[1].

Доказательство осуществляется следующим образом: После того, как Боб заходит в пещеру, Алиса, переместившись в коридор, может попросить его выйти либо с левой, либо с правой стороны. Этот шаг повторяется некоторое число раз. Не зная кода, Боб может попробовать угадать и выйти с правильной стороны с вероятностью 50%. Таким образом, каждый успешно выполненный Бобом запрос Алисы повышает вероятность того, что он знает код от двери C.

Происходящее подчеркивает важные аспекты доказательства с нулевым разглашением:

1. **Итеративность.** Предполагается, что в доказательство входит много итераций.
2. **Наличие случайного параметра.** В каждой итерации реализуется случайный параметр — в данном случае это сторона, с которой Боб должен выйти. Боб не знает значения параметра до начала соответствующей итерации.
3. **Вероятностный характер доказательства.**

Более реальным примером доказательства с нулевым разглашением является использование **гамильтонова цикла**.

6. Гамильтонов цикл

Пусть у Алисы имеется некоторый достаточно большой граф G, в котором существует известный ей гамильтонов цикл.

Гамильтонов цикл — замкнутый цикл, обходящий все вершины графа. Поиск такого цикла является NP-полной задачей.

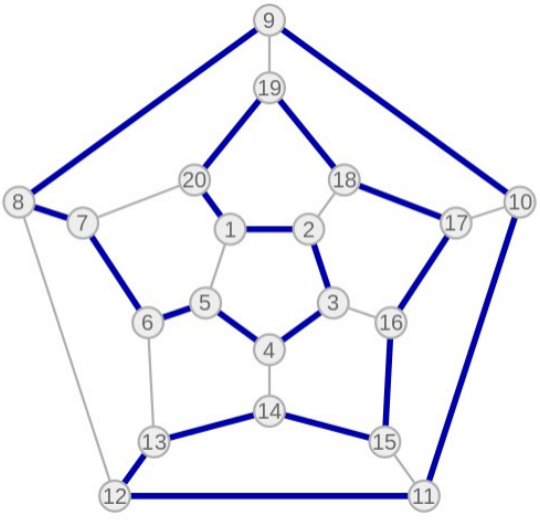

Рис. 11.2

[1] Алиса и Боб поменяны местами относительно общего случая в соответствии с объяснением лектора - *прим. ред.*

Пусть Алисе необходимо доказать Бобу свое знание такого цикла с нулевым разглашением. Для этого Алиса создает новый граф H, изоморфный G (перенумеровывает вершины G и запоминает соответствие), и передает его Бобу.

Далее Боб может попросить ее выслать ему одно из двух:

1. Соответствие $G \leftrightarrow H$ вершин G вершинам H.
2. Гамильтонов цикл в H.

При незнании G тем более невозможно указать гамильтонов цикл в H. Если Алиса мошенничает и знает гамильтонов цикл в некотором изоморфном G графе, то не сможет составить соответствие $G \leftrightarrow H$, т. к. при большом объеме графа его восстановление также является сложной задачей. Здесь вероятность угадывания на одной итерации та же, что и в предыдущем примере (50%), ведь Алиса может заранее подтасовать данные: либо передать Бобу некий изоморфный G граф, не зная цикла, либо передать произвольный граф, в котором она цикл знает. Нетрудно видеть, что данный механизм действительно обладает полнотой, корректностью и нулевым разглашением.

У представленного метода имеются следующие недостатки:

1. **Обман с несколькими личностями**
   Заранее придумав нужный замкнутый цикл, легко получить граф, в котором он будет гамильтоновым, и в результате появляется возможность быстро составить новые личности — предметы того, что можно доказать. По этой причине такого доказательства с нулевым разглашением не будет достаточно для авторизации — как минимум придется передать еще и имя пользователя.
2. **Обман, осуществляемый мафией**
   Благодаря посредническому общению вида Алиса → Мафия → Боб, мафия имеет возможность выдать себя за Алису перед Бобом. В результате она способна выполнить от ее имени, например, банковскую операцию, потребовав у нее доказательство с нулевым разглашением для оплаты счета в своем ресторане.
3. **Проблема гроссмейстера**
   Пусть Боб хочет показать Алисе, что является гроссмейстером мирового уровня[2]. Боб заранее договаривается об игре со вторым гроссмейстером, Евой, и играет, переключаясь между двумя партиями. В игре с Евой он использует ходы Алисы и наоборот. В результате он доказывает Алисе и Еве, что является гроссмейстером мирового уровня.

7. Слепая подпись

Пусть возникает необходимость подписать некоторый документ, не разглашая его содержимое заверяющему лицу, т. е. фактически не предъявляя самого документа.

Такое может потребоваться в случае с цифровыми валютами (с центральной эмиссией). Предположим, банк должен подписать платежный документ о перечислении средств

[2] Алиса и Боб снова поменяны местами в соответствии с объяснением лектора - *прим. ред.*

от одного абонента к другому, причем сумма сделки и прочие платежные данные должны остаться скрытыми от него.

Схожая задача может также возникает на избирательном участке. На выборах член счетной комиссии должен заверить голос избирателя, при этом не узнав, за кого он отдан.

Иллюстрации работы слепой подписи для криптосистемы RSA:

1. У Боба имеется открытая экспонента e и модуль n. Алисе нужно, чтобы он подписал число m без приобретения какой-либо информации о нем.

2. Алиса генерирует случайное число k и маскирует m:

$$t = m \cdot k^e \mod n.$$

3. Боб подписывает t:

$$u = t^d = (m \cdot k^e)^d \mod n = m^d \cdot k^{ed} \mod n$$

и отправляет Алисе.

4. Алиса производит снятие маскировки:

$$v = \frac{t^d}{k} = m^d \mod n.$$

8. Подбрасывание монетки по телефону

Допустим, Алисе и Бобу нужно решить некоторую проблему при помощи случайного числа. Это число должно быть соответствующим образом сгенерировано и защищено от нечестных ходов со стороны обоих абонентов. Например, если Алиса и Боб получат каждый по случайному биту и договорятся, что результатом будет число, полученое их побитовым сложением, легко устроить фальсификацию. Действительно, пусть, например, сначала Алиса получает бит Боба. В таком случае она сможет быстро подменить свой бит перед отправкой так, чтобы конечный результат был ей выгоден. Проблема, в итоге, состоит в неосуществимости одновременной передачи информации обоими абонентами.

Решение ее таково:

1. Алиса и Боб выбирают p, g.

2. Алиса выбирает

$$0 < x < p,$$

затем вычисляет

$$y = g^x \mod p$$

и посылает Бобу.

3. Боб выбирает случайное k и шифрует свой бит b

$$r = y^b \cdot g^k \mod p,$$

после чего отправляет Алисе.

4. Алиса выбирает свой бит c, о чем сообщает Бобу.

5. Боб посылает Алисе b и k.

6. Алиса проверяет честность Боба:

$$r \stackrel{?}{=} y^b \cdot g^k \mod p.$$

7. Результатом будет

$$b \oplus c.$$

Боб имеет возможность смухлевать, если найдет k_1 и k_2 для произвольной интерпретации отосланного r после сообщения Алисой свого бита.

$$r = 1 \cdot g^{k_1} \mod p$$

$$r = y \cdot g^{k_2} \mod p$$

Откуда следует, что

$$y = g^{k_1 - k_2} \mod p.$$

А это означает, что Боб умеет решать задачу дискретного логарифмирования, на сложности которой построена вся асимметричная криптография.