Криптографические протоколы

А.М. Миронов

Содержание

Глава 1

Задачи криптографии

1.1 Понятие криптографического протокола

Криптография является научной дисциплиной, изучающей математические модели безопасной передачи, обработки и хранения информации в небезопасной среде.

Криптографические методы защиты информации заключаются в применении для выполнения операций с защищаемой информацией специальных алгоритмов, которые называются **криптографическими протоколами**.

Криптографический протокол (КП) – это распределённый алгоритм, состоящий из нескольких **участников**, которыми могут быть

- люди
- компьютерные программы
- компьютеры и вычислительные комплексы
- базы данных
- сети связи
- банковские карточки и банкоматы
- и т.д.

Каждый из участников КП функционирует в соответствии с некоторым последовательным алгоритмом. Действия, исполняемые каждым из участников, могут иметь следующий вид:

- **посылка сообщения** другому участнику (или группе участников),
- **приём сообщения** от другого участника,
- **внутреннее действие**, которое заключается в выполнении участником некоторых вычислений.

1.1.1 Шифрование сообщений

Некоторые из сообщений, пересылаемых участниками КП, могут быть зашифрованными. Шифрование сообщений делается для того, чтобы противник, которому станут доступны пересылаемые сообщения, не смог ознакомиться с их содержанием.

Для шифрования сообщений могут использоваться различные системы шифрования, которые делятся на два класса: симметричные и асимметричные.

1. В **симметричных системах шифрования** для операций шифрования и расшифрования используется один и тот же ключ. Если K – ключ симметричной системы шифрования, то

   - для каждого сообщения m знакосочетание

     $$K(m)$$

     обозначает результат шифрования сообщения m на ключе K,

   - для каждого шифртекста m', зашифрованного на ключе K, знакосочетание

     $$K^{-1}(m')$$

     обозначает результат расшифрования этого шифртекста, т.е. для каждого сообщения m имеет место равенство

     $$K^{-1}(K(m)) = m$$

   Ниже знакосочетание вида K_{AB} (где A и B – некоторые участники) обозначает ключ некоторой симметричной системы шифрования, который известен участникам A и B.

2. В **асимметричных системах шифрования** для операций шифрования и расшифрования используются разные ключи. Ниже знакосочетания вида K_A^+ и K_A^- обозначают соответственно

   - открытый (который известен всем), и
   - закрытый (который известен только участнику A)

   ключи участника A в некоторой асимметричной системе шифрования.

   Для каждого сообщения m знакосочетание

   $$K_A^+(m)$$

   обозначает результат шифрования сообщения m на ключе K_A^+.

Для каждого шифртекста m', представляющего собой результат шифрования некоторого сообщения на ключе K_A^+, знакосочетание

$$K_A^-(m')$$

обозначает результат дешифрования этого шифртекста m' на ключе K_A^-, т.е. для каждого сообщения m имеет место равенство

$$K_A^-(K_A^+(m)) = m$$

В некоторых асимметричных системах шифрования для каждого сообщения m имеет место и другое равенство:

$$K_A^+(K_A^-(m)) = m$$

Системы шифрования с таким свойством используются в протоколах электронной подписи.

1.1.2 Классы участников КП

Участники, входящие в КП, делятся на три класса.

1. **Обычные (законные) участники** (обозначаются символами $A, B, \ldots$, возможно, с индексами).

2. **Доверенные посредники** (обозначаются символом T, возможно, с индексами).

3. **Противники**, которые делятся на следующие два класса.

   (a) **Пассивные противники** (обозначаются символом E, возможно, с индексами).

   Пассивный противник может перехватывать сообщения, пересылаемые другими участниками, и анализировать их.

   (b) **Активные противники** (обозначаются символом M, возможно, с индексами).

   Активный противник может

   - перехватывать сообщения, пересылаемые другими участниками, и анализировать их,
   - модифицировать или удалять пересылаемые сообщения
   - генерировать новые сообщения и посылать их другим участникам,
   - выдавать себя за другого участника (таких активных противников называют **мошенниками**).

1.1.3 КП безопасного взаимодействия в открытых сетях связи

Одной из задач, для решения которой могут использоваться КП, является задача организации безопасного взаимодействия в небезопасных открытых компьютерных сетях и распределённых компьютерных системах. Главные задачи, для решения которых используются такие КП, заключаются в следующем.

1. Аутентификация участников взаимодействия.
2. Распределение криптографических ключей.
3. Целостность и конфиденциальность передаваемых сообщений.
4. Сохранение порядка и своевременность передаваемых сообщений.

1.2 Формальное описание КП

Одним из простейших видов формального описания КП является перечисление записей вида

$$A_i \to A_j : m \tag{1.1}$$

где

- A_i и A_j – имена участников, и
- m – некоторое выражение, значением которого является сообщение.

Данные записи изображают действия, которые должны выполнять участники в процессе функционирования КП. Последовательность действий, обозначаемая знакосочетанием (1.1), имеет следующий вид:

- участник A_i
  - вычисляет значение выражения m, и
  - посылает его участнику A_j
- участник A_j
  - получает это сообщение, и
  - помещает его в определённое место своей локальной памяти.

Действия, входящие в список из знакосочетаний вида (1.1), исполняются последовательно. Если текущее исполняемое действие не относится к какому-либо из участников, то этот участник не функционирует в момент исполнения этого действия.

Отметим, что сообщением принято называть любой объект, который один из участников КП передаёт другому в процессе функционирования КП. Этот объект может быть не только текстовым сообщением, а, например, набором банкнот, товаром, и т.п.

Выражение m может начинаться с проверки некоторого условия φ, и в этом случае m имеет вид

$$\varphi \ ? \ m'$$

Условие φ представляет собой выражение, значением которого является 0 или 1. Исполнение действий вида

$$A_i \to A_j : \varphi \ ? \ m' \tag{1.2}$$

отличается от вышеизложенного тем, что

- сначала вычисляется значение выражения φ, и
- если оно равно 0, то на этом выполнение действий, связанных с записью (1.2), завершается, и происходит переход к следующей записи,
- а если оно равно 1, то работа происходит обычным образом, т.е A_i вычисляет значение m' и пересылает его A_j, а A_j принимает его и записывает в свою память.

1.3 Примеры КП

1.3.1 КП продажи компьютера

У A есть компьютер, а у B есть деньги. B хочет купить у A компьютер. Действия для достижения данной цели могут иметь следующий вид:

1. $A \to B$: компьютер
2. $B \to A$: деньги

Поскольку A и B не верят друг другу, то

- A хочет, чтобы сначала было 2, а затем 1,
- B хочет, чтобы сначала было 1, а затем 2.

Одним из возможных решений данной проблемы может быть КП, в котором, помимо A и B, принимает участие доверенный посредник T. Искомый КП может иметь следующий вид:

- $A \to T$: компьютер
- $B \to A$: деньги
- $A \to T$: подтверждение или опровержение того, что
  - полученная от B сумма соответствует стоимости компьютера, и
  - деньги не фальшивые
- $T \to B$: (от A поступило подтверждение) ? компьютер
- $T \to A$: (от A поступило опровержение) ? компьютер

Данный КП является приемлемым для A и B по следующим причинам:

- A верит, что
  - до окончания проверки денег T не передаст компьютер участнику B
  - T вернёт компьютер A, если B передаст A недостаточную сумму или фальшивые деньги
- B верит, что
  - пока A не послал T подтверждение, компьютер находится у T
  - сразу после того, как A послал T подтверждение T передаст B компьютер.

1.3.2 Обедающие криптографы

Рассмотрим следующую задачу.

За круглым столом сидят три криптографа и обедают. После того, как они пообедали и хотят заплатить, официант сообщает им, что их обед полностью оплачен, но не уточняет, кто именно платил.

Возможен один из двух вариантов:

1. либо обед оплатил один из криптографов,
2. либо за обед заплатила ФСБ.

Криптографы хотят выяснить, какой именно из вариантов имеет место, причём, если имеет место первый вариант, то те участники, которые не платили, не должны узнать, кто же конкретно заплатил.

Для решения этой задачи предлагается следующий КП.

Поскольку участники сидят за круглым столом, то каждая пара соседей может подбрасывать монету между собой, так, чтобы результат был известен только им двоим.

После того, как все три пары подбросили монету, каждый участник знает результаты двух подбрасываний (решка или орёл). Эти результаты могут быть

- либо одинаковыми (т.е. оба раза была решка, или оба раза был орёл),
- либо разными (т.е. при одном подбрасывании была решка, а при другом - орёл).

Каждый участник говорит другим "одинаково" или "по разному", причём тот, кто заплатил, утверждает противоположное (т.е. если надо сказать "одинаково" то он говорит "по-разному", и наоборот).

Если число ответов "по разному" чётно, то это значит, что обед оплатила ФСБ, иначе - один из них.

1.3.3 Подтверждение приёма

Рассмотрим следующую задачу: участники A и B используют симметричную шифрсистему для шифрования сообщений, причём для контроля правильности передачи каждое получаемое сообщение отсылается назад отправителю (чтобы отправитель был уверен, что сообщение получено в неискажённом виде).

Участники используют для этой цели следующий КП.

1. $A \to B : K_B^+(K_A^-(m))$
2. $B \to A : K_A^+(K_B^-(m))$
   (полученное сообщение возвращается как подтверждение приёма)

К сожалению, данный КП уязвим к следующей атаке: активный противник M может перехватить первое сообщение, и

- $M \to B : K_B^+(K_A^-(m))$
- $B \to M : K_M^+(K_B^-(K_M^+(K_A^-(m))))$

В итоге M получает m.

1.3.4 Вычисление среднего значения

Участники $A_1, \ldots, A_n$ имеют числа $a_1, \ldots, a_n$ соответственно. Они хотят вычислить $\frac{1}{n}\sum_{i=1}^{n} a_i$, причём каждый участник A_i не хочет разглашать своё число a_i.

Данную задачу можно решить при помощи следующего КП.

1. $A_1 \to A_2 : K^+_{A_2}(a_1 + N)$, где $N \underset{r}{\in} \mathbf{Z}$
   (символ "$\underset{r}{\in}$" означает случайный выбор элемента, т.е. знакосочетание $x \underset{r}{\in} X$ означает, что x является случайно и равновероятно выбранным элементом множества X)
2. $A_2 \to A_3 : K^+_{A_3}(a_2 + a_1 + N)$
3. $A_3 \to A_4 : K^+_{A_4}(a_3 + a_2 + a_1 + N)$
4. ...
5. $A_n \to A_1 : K^+_{A_1}(a_n + \ldots + a_1 + N)$
6. A_1 вычитает N из полученного результата и делит на n.

1.3.5 Сравнение двух чисел

Данный КП предназначен для решения следующей задачи: участники A и B имеют числа a и b соответственно. Они хотят узнать без разглашения чисел a и b, верно ли, что $a \leq b$.

Излагаемый ниже КП принадлежит Яо.

Мы будем предполагать, что $a, b \in \{1, \ldots, 100\}$.

1. $A \to B : c - a$, где $c = K^+_B(x)$ и $x \underset{r}{\in} \mathbf{Z}$
2. B генерирует простое p и вычисляет

$$\{c_i := K^-_B(c - a + i) \ \% \ p \mid i = 1, \ldots, 100\}$$

   B проверяет, что

   - $\forall i \neq j \quad |c_i - c_j| \geq 2$
   - $\forall i \quad 0 < c_i < p - 1$

   если это не выполняется, то шаг повторяется (с другим p)
3. $B \to A : \ c_1, \ldots, c_b, c_{b+1} + 1, \ldots, c_{100} + 1, p$
   (все эти числа - различны)
   если $c_i = c_j$, то $i \leq b < j$, и A узнает это
4. $A \to B$: ответ $= (a \leq b)$, если $x \underset{p}{=} a$-й член этой последовательности

Дырка: нет контроля честности участников.

Этот КП можно использовать для построения КП тайного аукциона, целью которого является решение следующей задачи:

- каждый из участников предлагает свою цену за некоторый предмет, и
- этот предмет достаётся победителю, которым считается участник, предложивший
  - максимальную цену (голландская система), или
  - цену, вторую после максимальной (английская система)

Чтобы не дать возможности участникам изменять в процессе аукциона уже сделанные предложения, надо использовать bit committment.

1.4 Свойства КП

Свойства КП подразделяются на несколько классов. Наибольшую актуальность представляют свойства из следующих классов.

1. **Корректность**, т.е
   - правильность вычислений, производимых участниками КП,
   - соответствие результатов, вычисленных участниками, заданным соотношениям
   - и т.п.

   Свойства корректности являются главными, так как в случае их нарушения использование КП невозможно, даже если этот КП обладает всеми остальными свойствами.

2. **Безопасность**.

   Данный класс свойств подразделяется на несколько подклассов, наиболее актуальными из которых являются следующие.
   - **Целостность**, которая заключается в том, что всякая попытка противника внести изменения в сообщения, которыми обмениваются законные участники, будет обнаружена в ходе выполнения КП.
   - **Секретность**, которая заключается в отсутствии неавторизованной утечки информации в процессе работы КП: при любом функционировании КП противник не должен быть способен ознакомиться с содержанием зашифрованных сообщений.

3. **Устойчивость** КП в случае
   - отклонения поведения участников от заданного поведения, и

- в случае непредусмотренного поведения окружения, в котором выполняется КП.

Также к данному классу свойств относится способность к быстрому восстановлению нормальной работы КП после сбоя компьютерной системы, на которой выполняется КП.

4. **Эффективность** использования ресурсов времени и памяти в процессе работы КП. Оптимальность алгоритмов, реализованных в КП.

5. **Адаптируемость** КП к небольшим изменениям окружения путём изменения его настроек, без изменения его внутренней структуры.

6. Чёткая и понятная **документированность** описания КП, позволяющая быстро модифицировать КП в случае существенного изменения условий его использования (например в случае расширения или сужения множества допустимых входных данных).

7. **Переносимость** и **совместимость**, т.е. способность КП одинаково хорошо работать на разных платформах и в разных конфигурациях.

1.5 Уязвимости в КП

Существует множество примеров опубликованных простых КП, которые, на первый взгляд, решали поставленные перед ними задачи, но в результате их использования выяснилось, что они содержат уязвимости (**flaws**).

Рассмотрим в качестве примера уязвимость в КП Нидхема - Шредера (Needham - Schroeder, 1979), который ниже мы будем обозначать знакосочетанием NS. Целью данного КП является

- аутентификация участников A и B друг перед другом, и
- совместная выработка ими общего сеансового ключа K_{AB} симметричного шифрования

Предполагается, что участники A и B используют общую систему асимметричного шифрования, и

- их открытые ключи имеют вид K_A^+ и K_B^+ соответственно, и
- их закрытые ключи имеют вид K_A^- и K_B^- соответственно.

Мы будем использовать следующее соглашение: если

- m – некоторое сообщение, и
- символ K обозначает некоторый ключ шифрования

то шифртекст, получаемый путём шифрования m на ключе K, будет обозначаться знакосочетанием $K(m)$.

В нашей системе обозначений КП NS представляется следующим образом:

1. $A \to B : K_B^+(Id_A, N_A)$
   где сообщение (A, N_A) состоит из двух компонентов:

   - Id_A есть идентификатор участника A, он говорит участнику B, кто является автором этого сообщения,
     (ниже мы будем обозначать идентификатор участника тем же символом, которым обозначается сам участник, т.е. вместо Id_A будем писать просто A)
   - N_A представляет собой битовую строку, случайным образом сгенерированную участником A
     (строки такого типа называются **нонсами**)

2. $B \to A : K_A^+(N_A, N_B)$
   где сообщение (N_A, N_B) состоит из двух компонентов:

   - N_A представляет собой нонс, который B извлёк из сообщения, полученного от A,
     B посылает A этот нонс для того, чтобы доказать участнику A, что если он смог извлечь N_A, то, следовательно, он знает K_B^- (который кроме него по предположению никто знать не может), и, следовательно, он на самом деле является B
   - N_B представляет собой нонс, сгенерированный участником B, который предназначен для того, чтобы A тоже получил возможность убедить B в своей подлинности

3. $A \to B : K_B^+(N_B)$
   когда B получает это сообщение, расшифровывает его, и сравнивает полученный открытый текст с тем нонсом N_B, который он послал A в зашифрованном виде, он убеждается в том, что участник, приславший ему это сообщение, действительно является A, так как, если он смог извлечь нонс N_B из шифртекста $K_A^+(N_A, N_B)$, то он знает закрытый ключ K_A^-, который кроме A никто знать не может.

После того, как участники A и B удостоверились в подлинности друг друга, в качестве сеансового ключа они могут выбрать, например ключ K_{AB}, получаемый побитовым сложением по модулю 2 строк N_A и N_B.

Одна из уязвимостей данного КП связана с тем, что один из законных участников (например, B), может использовать свой статус законного участника для того, чтобы выдавать себя за другого участника. Работа КП NS в этом случае может иметь следующий вид.

1. После того, как участник A выразит желание взаимодействовать с B по КП NS, и пришлёт ему первое сообщение (т.е. $K_B^+(A, N_A)$), участник B может использовать это сообщение для того, чтобы

   - взаимодействовать по КП NS с ещё одним участником C
   - и в этом взаимодействии B будет позиционировать себя как участник A.

   Для этого B посылает C сообщение $K_C^+(A, N_A)$.

2. Получив это сообщение и расшифровав его, C делает вывод, что это сообщение было послано участником A (о чём говорит ему первая компонента расшифрованного сообщения).

   На основании этого, C

   - генерирует нонс N_C, и
   - посылает участнику B шифртекст

     $$K_A^+(N_A, N_C)$$

     (думая, что он посылает его участнику A)

3. B пересылает полученное от C сообщение участнику A.

4. Участник A рассматривает полученное от B сообщение как на то сообщение, которое B должен послать ему в соответствии со вторым шагом КП NS, т.е. A рассматривает извлечённую из полученного шифртекста компоненту N_C как нонс, который был сгенерирован участником B.

5. В соответствии с третьим шагом КП NS, A посылает B шифртекст $K_B^+(N_C)$.

6. B извлекает N_C и посылает участнику C шифртекст $K_C^+(N_C)$.

После этого B знает всю информацию (а именно, нонсы N_A и N_C), которая позволит ему создать сеансовый ключ для шифрованного взаимодействия с C, в котором он будет выступать от имени A.

Описанный выше пример уязвимости, а также многие другие примеры уязвимостей в КП являются убедительным свидетельством того, что для практических КП, используемых в критических по безопасности системах, недостаточно одного лишь визуального анализа разработанного КП, и для гарантированного обоснования надёжности КП необходимо использовать автоматизированные методы анализа КП.

1.6 Предположения о противнике

Как правило, исчерпывающего решения задачи обоснования того, что КП обладает заданными свойствами, получить невозможно, поэтому обычно ограничиваются приближённым решением этой задачи, когда

- противник, атакующий КП, предполагается не произвольным, а принадлежащим некоторому заданному классу
- корректность КП обосновывается в предположении стойкости криптографических примитивов, используемых в КП, и
- утверждение о корректности КП сопровождается некоторым параметром или модальностью, выражающим (количественно или качественно) уровень доверия к этому утверждению.

При решении задач анализа КП как правило используются следующие предположения о противнике:

- противник полностью знает КП,
- противнику недоступна информация, которую участники хранят в своих локальных ресурсах памяти
  (в частности, противнику недоступны все секретные ключи, которые невозможно вычислить аналитически)
- противнику доступны все пересылаемые сообщения между участниками, которые он может
  - модифицировать
  - удалять
  - заменять своими сообщениями
- противник не может скомпрометировать используемые в КП **криптографические примитивы,** к которым относятся
  - алгоритмы шифрования и дешифрования
  - алгоритмы электронной подписи
  - хэш-функции
  - генераторы псевдослучайных чисел, и т.д.

Глава 2

Системы шифрования

2.1 Симметричные системы шифрования

Симметричные системы шифрования в основном относятся к следующим двум классам: блочные и поточные.

2.1.1 Блочные системы шифрования

В блочных системах шифрования открытый текст перед шифрованием разбивается на блоки, состоящие из нескольких битов, и каждый блок шифруется при помощи одного и того же алгоритма.

Шифрующие преобразования блоков заключаются в суперпозиции нескольких простых некоммутирующих отображений, называемых **базовыми преобразованиями**.

Среди базовых преобразований наибольшее распространение получили **преобразования Файштеля**, которые заключаются в

- разделении обрабатываемого блока на левую и правую половины L и R, и
- преобразовании блока (L, R) в блок (L', R') (где L' и R' – левая и правая половины преобразованного блока) по следующему принципу:

$$\begin{cases} L' := R \\ R' := L + f(R, K) \end{cases}$$

где

  - символ + обозначает побитовое сложение по модулю 2
  - K обозначает битовый вектор, называемый **ключом**
  - f обозначает некоторую булеву функцию.

Преобразование Файштеля обратимо, и обратное преобразование вычисляется по той же схеме, как и исходное:

$$\begin{cases} L := R' + f(L', K) \\ R := L' \end{cases}$$

Алгоритм шифрования реализуется несколькими итерациями преобразования Файштеля, при этом очередная (i–я) итерация использует в качестве входного блока (L_i, R_i) результат предыдущей итерации.

2.1.2 Поточные системы шифрования

В поточных системах шифрования каждый бит открытого текста шифруется отдельно. Как правило, шифрование заключается в сложении по модулю 2 каждого бита открытого текста с соответствующим битом последовательности, вырабатываемой шифрующим устройством, построенным с использованием одного или нескольких регистров сдвига с линейной обратной связью (данная последовательность называется **управляющей последовательностью**).

Регистры сдвига с линейной обратной связью

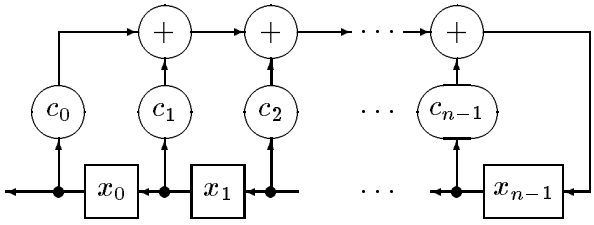
Ниже символ $\mathbf{B}$ обозначает поле из двух элементов 0 и 1.

Регистр сдвига с линейной обратной связью (РСЛОС, в англоязычной литературе используется аббревиатура **LFSR**) представляет собой устройство для порождения линейной рекуррентной последовательности булевых значений $b_0, b_1, \ldots$, удовлетворяющей соотношению: для каждого $i \geq 0$

$$b_{n+i} = c_{n-1}b_{n+i-1} + c_{n-2}b_{n+i-2} + \ldots + c_0 b_i$$

где $c_0, \ldots, c_{n-1}$ – некоторые фиксированные элементы поля $\mathbf{B}$.

Линейный регистр сдвига часто изображают диаграммой



РСЛОС можно представлять себе как автомат (S, δ, f), где

- $S = \mathbf{B}^n$ – множество состояний, состоит из всех булевых векторов длины n

- $\delta : S \to S$ – отображение перехода, имеющее следующий вид:

$$\delta(x_0, \ldots, x_{n-1}) = (x_1, \ldots, x_{n-1}, \sum_{i=0}^{n-1} c_i x_i)$$

- $f : S \to \mathbf{B}$ – отображение выхода, это булева функция (называемая функцией усложнения). Мы будем предполагать, что f представляет собой полином от переменных $x_0, \ldots, x_{n-1}$.

Узел выборки

Узел выборки - это криптографический примитив, генерирующий управляющую последовательность. Данный примитив построен из двух РСЛОС, выходы которых поступают на входы автомата, работа которого заключается в преобразовании двух входных последовательностей булевых значений

$$a_0, a_1, a_2, \ldots, \quad \text{и} \quad b_0, b_1, b_2, \ldots$$

в одну выходную последовательность

$$c_0, c_1, c_2, \ldots$$

получаемую из последовательности $a_0, a_1, a_2, \ldots$ удалением всех компонентов с такими номерами i, что $b_i = 0$.

2.2 Асимметричные системы шифрования

2.2.1 Система шифрования RSA

Параметры:

- p, q – простые числа, удовлетворяющие условиям:
  - p и q примерно одинаковы по размеру,
  - НОД $(p-1, q-1)$ – небольшое число
- $K^+ := \{n, e\}$, где $n := pq, \quad e \underset{r}{\in} \mathbf{Z}^*_{\varphi(n)}$.

  Отметим, что если
  - e – простое, и
  - $e > \max(p, q)$

  то $e \in \mathbf{Z}^*_{\varphi(n)}$, т.к. $\varphi(n) = (p-1)(q-1)$. Поэтому можно генерировать e с помощью тестов проверки на простоту.

- $K^- := \{d, p, q\}$, где $ed \underset{\varphi(n)}{=} 1$, т.е. d ищется как решение уравнения

$$e \cdot d = t \cdot \varphi(n) + 1$$

Блоки ОТ и ШТ представляются элементами $\mathbf{Z}_n$. Ниже все вычисления и сравнения - по $\mathrm{mod}\ n$.

1. Шифрование: $K^+(m) \stackrel{\text{def}}{=} m^e$.
2. Дешифрование: $K^-(u) \stackrel{\text{def}}{=} u^d$.

Докажем, что для каждого $x \in \mathbf{Z}_n$ имеет место равенство

$$(x^e)^d = x \tag{2.1}$$

- Если $x \in \mathbf{Z}_n^*$, то $x^{\varphi(n)} = 1$ (т.к. порядок группы $\mathbf{Z}_n^*$ равен $\varphi(n)$), поэтому

$$(x^e)^d = x^{ed} = x^{t\cdot\varphi(n)+1} = (x^{\varphi(n)})^t \cdot x = 1^t \cdot x = x$$

- Если $x \in \mathbf{Z}_n \setminus \mathbf{Z}_n^*$, то такой x имеет вид
  - pi (где $i = 0, \ldots, q-1$), или
  - qi (где $i = 0, \ldots, p-1$).

  Расмотрим, например, первый случай, т.е. $x = pi$, где $i = 0, \ldots, q-1$.

  Если $i = 0$, то $x = 0$, и равенство (2.1) верно.

  Если $i \neq 0$, то число pi взаимно просто с q. Разделим pi с остатком на q:

$$pi = qj + r, \qquad \text{где } r \in \mathbf{Z}_q^*$$

  Порядок группы $\mathbf{Z}_q^*$ равен $q-1$, поэтому

$$r^{q-1} \underset{q}{=} 1$$

  т.е. $(pi - qj)^{q-1} - 1 \underset{q}{=} 0$. Раскрывая $(pi - qj)^{q-1}$ по формуле бинома, получаем:

$$\begin{aligned}
&(pi)^{q-1} - 1 \underset{q}{=} 0 && \Rightarrow \\
&(pi)^{q-1} \underset{q}{=} 1 && \Rightarrow \\
&(pi)^{t\cdot(p-1)\cdot(q-1)} \underset{q}{=} 1 && \Rightarrow \\
&(pi)^{t\cdot\varphi(n)} \underset{q}{=} 1 && \Rightarrow \\
&((pi)^{t\cdot\varphi(n)} - 1) \underset{q}{=} 0 && \Rightarrow \\
&((pi)^{t\cdot\varphi(n)} - 1) = q \cdot k && \Rightarrow \\
&(pi) \cdot ((pi)^{t\cdot\varphi(n)} - 1) = pq \cdot ik && \Rightarrow \\
&x \cdot (x^{t\cdot\varphi(n)} - 1) \underset{n}{=} 0
\end{aligned}$$

откуда следует (2.1).

2.2.2 Система шифрования Эль-Гамаля

Параметры:

- p – простое число
  (ниже все вычисления и сравнения - по $\mathrm{mod}\ p$)
- $K^- := q \in \{1, \ldots, p-2\}$,
- $K^+ := \{p, \alpha, \beta := \alpha^q\}$, где $\alpha \in \mathbf{Z}_p^*$ – примитивный элемент

Шифрование:

$$K^+(m) := (\alpha^x,\ m \cdot \beta^x), \qquad \text{где } x \underset{r}{\in} \{1, \ldots, p-2\}$$

Дешифрование: $K^-(u, v) := u^{-q} \cdot v$.

Глава 3

Хэш-функции

3.1 Определение хэш-функции

Хэш-функция (ХФ) - это функция h вида

$$h : D \to \{0,1\}^m, \quad \text{где } D \subseteq \{0,1\}^*$$

которая обладает следующими свойствами:

1. h – **однонаправленная функция**, т.е. не существует быстрого алгоритма вычисления h^{-1}
2. h **устойчива к коллизиям**, т.е. сложно найти различные $u, v \in D$, такие, что $h(u) = h(v)$.

Из свойства 2 следует свойство, которое называют **устойчивостью к нахождению второго прообраза**: для заданного $u \in D$ сложно найти $v \neq u$, такой, что $h(u) = h(v)$.

ХФ могут использоваться для

- контроля целостности передаваемых или хранимых данных (для этих целей значение m м.б. от 32 до 64)
- аутентификации источника данных (для этих целей m должно быть ≥ 128)

Строку $h(u)$ иногда называют **имитовставкой**, или **кодом аутентификации** сообщения u.

3.2 Пример построения ХФ

ХФ может быть построена на основе одношаговой сжимающей функции

$$\delta : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}^m$$

Вычисление $h(u)$ производится следующим образом:

- строка u дополняется до размера, кратного m, и разбивается на блоки длины m:
$$u =: (u_1, \ldots, u_k)$$
- вычисляются строки $q_0, \ldots, q_k$, где
  - q_0 – некоторая фиксированная строка, и
  - $q_i := f(u_i, q_{i-1}) \quad (i = 1, \ldots, k)$
- $h(u) := q_k$.

Примеры хороших δ:

1. $\delta(x_1, x_2) := x_1(x_2) \oplus x_2$ (или $x_2(x_1) \oplus x_1$), где $x_i(x_j)$ обозначает ШТ, получаемый шифрованием ОТ x_j на ключе x_i в блоковой ШС.
2. $\delta(x_1, x_2) := G(x_2)(x_1) \oplus x_1$ где $G(x_2)(x_1)$ обозначает ШТ, получаемый шифрованием ОТ x_1 на ключе $G(x_2)$ в блоковой ШС
3. $\delta(x_1, x_2) := G(x_2)(x_1) \oplus x_1 \oplus x_2$, где все параметры такие же, как в предыдущем пункте.

Пример плохой δ:
$$\delta(x_1, x_2) := K(x_1 \oplus x_2)$$
где $K(x_1 \oplus x_2)$ обозначает ШТ, получаемый шифрованием ОТ $x_1 \oplus x_2$ на ключе K в блоковой ШС.

3.3 Стандарт ХФ SHS

В этом пункте мы рассмотрим американский стандарт ХФ SHS (Secure Hash Standard), он был разработан в 1993 г. Этот стандарт основан на алгоритме MD4 Ривеста. ХФ, построенная по данному стандарту, преобразует строки длины $\leq 2^{64}$ в строки длины 160.

Вычисление значения $h(a_1, \ldots, a_n)$ производится следующим образом. Сначала строка $(a_1, \ldots, a_n)$ дополняется до строки u, размер которой делится на 512, и которая имеет вид
$$u = (a_1, \ldots, a_n, 1, 0, \ldots, 0, l_1, \ldots, l_{64})$$
где $(l_1, \ldots, l_{64})$ – двоичная запись n. Разобьем строку u на блоки $u_1, \ldots, u_k$ длины 512. Искомое значение $h(a_1, \ldots, a_n)$ равно состоянию, в которое перейдёт автомат (U, Q, q_0, δ) после поступления на его вход последовательности $u_1, \ldots, u_k$, где

- $U = \{0,1\}^{512}$ – множество входных сигналов

- $Q = \{0,1\}^{160}$ – множество состояний
- $q_0 \in Q$ - начальное состояние
- $\delta : Q \times U \to Q$ – отображение переходов

Отображение переходов данного автомата определяется следующим образом:

$$\delta(q, x) := q + g(q, x)$$

где сложение понимается следующим образом: слагаемые разбиваются на 5 слов $\in \{0,1\}^{32}$, и соответствующие слова складываются по $\bmod 2^{32}$. Выражение $g(q, x)$ обозначает состояние, в которое перейдёт автомат

$$(U', Q', q, \delta'), \quad \text{где} \begin{cases} U' = \{0,1\}^{32} \\ Q' = (\{0,1\}^{32})^5 \\ q - \text{начальное состояние} \\ \delta' : Q' \times U' \to Q' - \\ \quad \text{отображение переходов} \end{cases}$$

после поступления на его вход последовательности $(v_0, \ldots, v_{79})$, которая определяется следующим образом:

- последовательность $(v_0, \ldots, v_{15})$ представляет собой разбиение x на 16 слов длины 32
- для каждого $i = 16, \ldots, 79$

$$v_i := v_{i-3} \oplus v_{i-8} \oplus v_{i-14} \oplus v_{i-16}$$

Отображение переходов δ' определяется следующим образом. Пусть

- текущее состояние автомата имеет вид

$$(a_i, b_i, c_i, d_i, e_i) \in (\{0,1\}^{32})^5$$

- входной сигнал в текущий момент равен v_i.

Тогда состояние $(a_{i+1}, b_{i+1}, c_{i+1}, d_{i+1}, e_{i+1})$ данного автомата в следующий момент времени определяется следующим образом:

- $a_{i+1} := (T^5(a_i) + f_i(b_i, c_i, d_i) + e_i + v_i + z_i) \ \% \ 2^{32}$
- $b_{i+1} := a_i$

- $c_{i+1} := T^{30}(b_i)$
- $d_{i+1} := c_i$
- $e_{i+1} := d_i$

где

- T – циклический сдвиг влево на 1 бит
- $$f_i(b,c,d) = \\ = \begin{cases} (b \wedge c) \vee (\overline{b} \wedge d) & (i = 0, \ldots, 19) \\ b \oplus c \oplus d & (i = 20, \ldots, 39, 60, \ldots, 79) \\ (b \wedge c) \vee (b \wedge d) \vee (c \wedge d) & (i = 40, \ldots, 59) \end{cases}$$
  где $\vee$ и $\wedge$ – побитовые дизъюнкция и конъюнкция векторов, и $\overline{b}$ – побитовое инвертирование
- $z_0, \ldots, z_{79}$ – 32-битовые слова, которые имеют один и тот же вид для
  - $i = 0, \ldots, 19$
  - $i = 20, \ldots, 39$
  - $i = 40, \ldots, 59$
  - $i = 60, \ldots, 79$

Глава 4

КП аутентификации

4.1 Понятие аутентификации

КП аутентификации (ПА) предназначены для решения задачи **аутентификации** (т.е. установления подлинности) участников КП.

Понятие аутентификации может относиться не только к участникам КП, но также и к

- сообщениям, пересылаемым в процессе работы КП,
- сеансам связи,
- времени создания сообщений, и т.д.

Ниже мы рассматриваем только вопросы аутентификации участников.

Проблема установления подлинности участников представляет большую актуальность, например, в том случае, когда участники выражают желание получить доступ к ресурсам, безопасность которых представляет повышенный интерес, таким, например, как

- банковские счета
- секретные базы данных
- государственные здания, и т.д.

Часто в результате работы ПА происходит выработка сеансового ключа симметричного шифрования, который участники ПА используют для организации сеанса шифрованной связи после завершения работы ПА.

Как правило, в ПА принимают участие два обычных участника, а также может принимать участие доверенный посредник. Протоколы аутентификации включают в себя следующие два класса:

1. КП **односторонней аутентификации**, в которых только один из участников доказывает свою подлинность другому участнику, и

2. КП **двусторонней аутентификации**, в которых оба участника доказывают свою подлинность друг другу.

Под доказательством подлинности в разных ПА могут пониматься разные вещи:

- в одних ПА в качестве таких доказательств выступает доказательство знания некоторого секрета, который может быть известен
  - только тому участнику, который доказывает свою подлинность, например закрытый ключ этого участника, или
  - только двум участникам:
    * участнику, который доказывает свою подлинность, и
    * участнику, который проверяет это доказательство

    в качестве такого секрета может выступать пароль или PIN,
- в других ПА (участниками которых являются живые люди) в качестве таких доказательств могут выступать
  - отпечатки пальцев,
  - характеристики глаза, голоса, и т.п.

4.2 КП односторонней аутентификации

В КП односторонней аутентификации лишь один из участников доказывает свою подлинность другому участнику. Работа такого КП обычно состоит из нескольких раундов, в каждом из которых

1. доказывающий участник (обозначаемый символом P, от слова Prover) предъявляет доказательство своей подлинности (т.е. что он действительно тот, в качестве кого он себя позиционирует) проверяющему участнику, и

2. проверяющий участник (обозначаемый символом V, от слова Verifier). проверяет доказательство, и принимает решение: признавать ли предъявленное доказательство правильным, или нет.

Например, в качестве P может выступать кредитная карточка, а в качестве V – банкомат.

Как правило, в каждом раунде

- участник V посылает участнику P некоторый вопрос, и
- сообщение, которое в этом раунде P посылает V в качестве доказательства своей подлинности, представляет собой ответ на этот вопрос.

P проходит аутентификацию только в том случае, если в каждом из раундов участник V признаёт правильным ответ, который ему послал V в этом раунде.

Для защиты от атаки с повторной передачей (**replay**) можно включать в каждый вопрос и ответ нонс (т.е. случайную строку) или метку времени. В этом случае при проверке ответа производится дополнительная проверка того, что

- нонс, содержащийся в ответе, совпадает с нонсом, содержащемся в вопросе, или
- метка времени в ответе принадлежит заданному промежутку $[t_{min}, t_{max}]$.

4.3 Однораундовые КП односторонней аутентификации

4.3.1 Простейшие примеры однораундовых КП односторонней аутентификации

1. с использованием симметричной шифрсистемы и метки времени:

$$P \to V : K_{PV}(t, V)$$

2. с использованием симметричной шифрсистемы и нонса:
   - $V \to P : N$
   - $P \to V : K_{PV}(N, V)$
3. с использованием электронной подписи и метки времени:

$$P \to V : S_P(t, V)$$

4. с использованием электронной подписи и нонса:
   - $V \to P : N$
   - $P \to V : S_P(N, V)$
5. с использованием асимметричной шифрсистемы и нонса:

- $V \to P : N$
- $P \to V : P, K_P^-(N)$

6. с использованием хэш-функции и нонсов:
   - $V \to P : N_V$
   - $P \to V : \ N_P, \ h(N_V, N_P, P)$
7. с использованием асимметричной шифрсистемы, хэш - функции и нонса
   - $V \to P : \ h(N), \ V, \ K_P^+(N, P)$
   - $P \to V : N$

4.3.2 ПА с использованием паролей

Пароль - это случайная строка из ≥ 9 символов, являющаяся общим секретом P и V. ПА с использованием паролей представляют собой однораундовые КП односторонней аутентификации. Класс ПА с использованием паролей делится на два подкласса:

- ПА с фиксированным паролем, и
- ПА с меняющимися паролями.

ПА с фиксированным паролем

ПА с фиксированным паролем состоит из пересылки одного сообщения, которое может иметь, например, такой вид:

$$P \to V : \ (P, \text{ пароль, желаемый ресурс})$$

Отметим, что для защиты от взлома паролей путём их перехвата или подбора можно использовать «подсоленные» пароли, которые получаются из обычных паролей добавлением большой случайной битовой строки (которая называется **солью**), и в этом случае участник P пересылает вместо пароля строку

$$h(\text{ пароль, соль})$$

где h – некоторая хэш-функция.

ПА с меняющимися паролями

В ПА данного типа пароли в разных сеансах аутентификации могут быть разными. Обновление паролей может происходить по одной из следующих схем.

1. Одна из таких схем заключается в том, что P и V имеют общий список паролей $\pi_1, \ldots, \pi_n$, и заранее договариваются о порядке смены паролей в КП аутентификации.

2. Другой схемой является выработка новых паролей по следующему алгоритму.

   Сначала P и V имеют один общий пароль π_0. Каждый сеанс аутентификации P и V имеет свой порядковый номер $i = 0, 1, \ldots$. На сеансе аутентификации с номером i используется пароль π_i. Пароли $\pi_1, \ldots$ генерируются участником P. Предполагается, что P и V используют алгоритм, который по каждому паролю π_i вырабатывает ключ шифрования K_{π_i}. i–й сеанс аутентификации P с V содержит пересылку

   $$P \to V : K_{\pi_i}(\pi_{i+1})$$

   в которой P посылает V пароль, который будет использоваться на следующем сеансе аутентификации.

3. Ещё одна схема использует хэш - функцию h.

   P и V выбирают число n, представляющее собой максимальное количество сеансов аутентификации, которые они собираются выполнить.

   P выбирает случайную строку N, и генерирует последовательность паролей

   $$\begin{aligned} \pi_0 &:= N \\ \pi_1 &:= h(N) \\ &\ldots \\ \pi_{n+1} &:= h^{n+1}(N) \end{aligned}$$

   V каким-либо образом получает π_{n+1}.

   i–й сеанс аутентификации (где $i = 1, \ldots, n$) содержит действие

   $$P \to V : \ P,\ i,\ \pi_{n-i+1}$$

   где символ π_{n-i+1} обозначает пароль, который используется в этом сеансе.

   В каждом сеансе аутентификации участник V

   - проверяет равенство $h(\pi) = \pi'$, где
     - π – пароль, который в этом сеансе ему прислал P, и
     - π' – пароль, который использовался в предыдущем сеансе (если $i = 1$, то $\pi' = \pi_{n+1}$).

- запоминает пароль π для использования его при проверке на следующем сеансе.

4. Можно определить ещё одну схему генерации последовательности паролей, в которой при генерации каждого нового пароля используется нонс, который использовался при генерации предыдущего пароля.

4.3.3 ПА Шнорра

В этом ПА участники P и V выбирают

- простые числа p и q, где
  - длина p примерно равна 512 бит
  - длина q примерно равна 140 бит
  - $p-1$ делится на q
- число $g \in \mathbf{Z}_p \setminus \{1\}$, такое, что $g^q \underset{p}{=} 1$

В качестве секрета, знание которого участник P доказывает участнику V, выступает число

$$s \underset{r}{\in} \mathbf{Z}_q^*$$

Предполагается, что V знает число $\nu := g^{-s} \pmod p$.

ПА Шнорра имеет следующий вид:

1. $P \to V : r := g^k \ \% \ p$, где $k \underset{r}{\in} \mathbf{Z}_q^*$
2. $V \to P : e \underset{r}{\in} \{0, \dots 2^t - 1\}$, где t примерно равно 72
3. $P \to V : s := (k + se) \ \% \ q$
4. V принимает ответ, если $r \underset{p}{=} g^s \nu^e$.

4.4 Многораундовые КП односторонней аутентификации

4.4.1 ПА Фиата - Шамира

В этом ПА в качестве секрета, знание которого участник P доказывает участнику V, выступает число

$$s \underset{r}{\in} \{2, \dots, n-2\}$$

где

- n – открытое число вида pq, где p, q – секретные простые числа длины ≥ 512 бит,
- V знает число $\nu := s^2$.

ПА состоит из t раундов, каждый из которых имеет следующий вид: (ниже все операции и сравнения выполняются по $\bmod n$)

1. $P \to V : x := z^2$, где $z \underset{r}{\in} \{1, \ldots, n-1\}$
2. $V \to P : b$, где $b \underset{r}{\in} \{0, 1\}$ (b – это вопрос)
3. $P \to V : \ y := \ z \cdot s^b$
   (отметим, что P не раскрывает значение s, т.к. число z – случайное, и никто, кроме P, его не знает)
4. V принимает ответ P, если $y^2 = \ x \cdot \nu^b$.

Вероятность того, что P, не зная s, успешно пройдёт аутентификацию, не превосходит 2^{-t}, так как

- если P знает s, то он правильно ответит на все вопросы, и
- если P не знает s, то вероятность успешного ответа P в одном раунде равна $1/2$, и, поскольку все раунды независимы, то вероятность того, что P не ошибётся во всех раундах, равна произведению вероятностей успешного ответа P в одном раунде, т.е. 2^{-t}.

Отметим, что даже если сообщения, которыми обмениваются P и V в процессе работы КП, будут перехвачены противником, из них невозможно извлечь никакой информации, о том, в какой области может лежать значение s. Это обосновывается сложностью вычисления функции $\sqrt{x}$ в $\mathbf{Z}_n$ для $n = pq$, где p и q – неизвестные простые числа (если бы p и q были известны, то вычислить эту функцию было бы просто).

ПА такого типа представляют собой **доказательства с нулевым разглашением**, или, сокращённо, Д0Р.

4.4.2 ПА Фейге - Фиата - Шамира

Данный ПА является модификацией предыдущего ПА. В этом ПА в качестве секрета, знание которого участник P доказывает участнику V, выступает не одно число s, а список из k чисел

$$(s_1, \ldots, s_k) \underset{r}{\in} \{2, \ldots, n-2\}^k$$

где

- n – открытое число вида pq, где p, q – секретные простые числа длины ≥ 512 бит,
- V знает числа $\nu_1 := (s_1^2)^{-1}, \ldots, \nu_k := (s_k^2)^{-1}$,
- числа $\nu_1, \ldots, \nu_k$ должны быть различны.

Данный ПА тоже состоит из t раундов, каждый из которых имеет следующий вид: (ниже все операции и сравнения выполняются по mod n)

1. $P \to V : x := z^2$, где $z \underset{r}{\in} \{1, \ldots, n-1\}$
2. $V \to P : (b_1, \ldots, b_k) \underset{r}{\in} \{0,1\}^k$
3. $P \to V : y := z \cdot (s_1^{b_1} \cdot \ldots \cdot s_k^{b_k})$
4. V принимает ответ P, если

$$x = y^2 \cdot (\nu_1^{b_1} \cdot \ldots \cdot \nu_k^{b_k})$$

Вероятность того, что P, не зная список $(s_1, \ldots, s_k)$, успешно пройдёт аутентификацию, равна 2^{-kt}.

Рекомендуемые значения для k и t: $k = 5$, $t = 4$.

4.4.3 ПА Гиллу - Кискате

Данный ПА представляет собой обобщение ПА Фиата - Шамира. Он тоже является Д0Р.

В этом ПА в качестве секрета, знание которого участник P доказывает участнику V, выступает число

$$s \underset{r}{\in} \{2, \ldots, n-2\}$$

где

- n – открытое число вида pq, где p, q – секретные простые числа длины ≥ 512 бит,
- V знает число

$$\nu := (s^k)^{-1} \pmod{n}$$

где $k \geq 2$ – некоторое открытое число.

ПА состоит из t раундов (где t - невелико, поэтому данный ПА можно использовать, например, в смарт - картах). Каждый раунд имеет следующий вид: (ниже все операции и сравнения выполняются по mod n)

1. $P \to V : x := z^k$, где $z \underset{r}{\in} \{1, \ldots, n-1\}$
2. $V \to P : b \underset{r}{\in} \{0, \ldots, k-1\}$
3. $P \to V : y := z \cdot s^b$
4. V принимает ответ P, если $x = y^k \cdot \nu^b$.

Корректность этого КП следует из равенств

$$y^k \cdot \nu^b = (z \cdot s^b)^k \cdot \nu^b = z^k \cdot (\nu \cdot s^k)^b = z^k$$

4.5 КП двусторонней аутентификации

В этом пункте мы приведём несколько простейших примеров КП двусторонней аутентификации.

1. с использованием симметричной шифрсистемы и нонсов
   - $B \to A : N_B$
   - $A \to B : K_{AB}(N_A, N_B, B)$
   - $B \to A : K_{AB}(N_A, N_B)$
2. с использованием электронной подписи и нонсов
   - $B \to A : N_B$
   - $A \to B : S_A(N_A, N_B, B)$
   - $B \to A : S_B(N_B, N_A, A)$
3. с использованием хэш - функции и нонсов
   - $B \to A : N_B$
   - $A \to B : \ N_A, \ h(A, N_A, N_B)$
   - $B \to A : h(N_A, B)$

4.6 КП аутентификации и передачи сеансовых ключей

Как уже было сказано выше, некоторые КП предназначены для одновременного решения двух задач:

- аутентификации участников, и
- передачи новых сеансовых ключей.

В этой секции мы рассмотрим наиболее важные КП такого вида.

4.6.1 Односторонняя аутентификация и передача сеансового ключа

В этом пункте мы представляем три КП односторонней аутентификации и передачи сеансового ключа с использованием

- асимметричной шифрсистемы,
- электронной подписи, и
- метки времени

Все эти КП состоят из одной пересылки сообщения.

1. $A \to B : K_B^+ \left(\ S_A(B, K_{AB}, t) \ \right)$
2. $A \to B : S_A \left(\begin{array}{l} B \\ t \\ K_B^+(A, K_{AB}) \end{array} \right)$
3. $A \to B : \left\{ \begin{array}{l} K_B^+(K_{AB}, t) \\ S_A(B, K_{AB}, t) \end{array} \right.$

4.6.2 Wide Mouth Frog

1. $A \to T : \left\{ \begin{array}{l} A \\ K_{AT}(t_A, B, K_{AB}) \end{array} \right.$
2. $T \to B : K_{BT}(t_B, A, K_{AB})$

4.6.3 Передача сеансового ключа с односторонней аутентификацией

1. $B \to A : N_B$
2. $A \to B : K_{AB}(K'_{AB}, N_B, B)$

где K_{AB} и K'_{AB} – соответственно старый и новый сеансовые ключи.

Вторая пересылка может иметь и такой вид:

$$A \to B : K'_{AB} \oplus h_{K_{AB}}(N_B, B) \tag{4.1}$$

где h – ключевая хэш-функция.

4.6.4 Передача сеансового ключа с двусторонней аутентификацией

1. (a) $A \to B : N_A$
   (b) $B \to A : K_{AB}(N_A, N_B)$
   (c) $A \to B : K_{AB}(K'_{AB}, N_B, B)$

   где K_{AB} и K'_{AB} – соответственно старый и новый сеансовые ключи.

   Вторая пересылка может иметь вид (4.1).

2. с использованием меток времени, электронной подписи, асимметричного и симметричного шифрования

   (a) $A \to B : \begin{cases} S_A(K_B^+(K_{AB})) \\ K_{AB}(t_A) \end{cases}$

   (b) $B \to A : K_{AB}(t_B)$

4.6.5 Otway-Rees

1. $A \to B : \begin{cases} N, A, B, \\ K_{AT}(N_A, N, A, B) \end{cases}$

2. $B \to T : \begin{cases} N, A, B, K_{AT}(N_A, N, A, B) \\ K_{BT}(N_B, N, A, B) \end{cases}$

3. $T \to B : \begin{cases} N \\ K_{AT}(N_A, K_{AB}) \\ K_{BT}(N_B, K_{AB}) \end{cases}$

4. $B \to A : \begin{cases} N \\ K_{AT}(N_A, K_{AB}) \end{cases}$

Если заменить шаг 4 на следующие два шага:

- $A \to B : \begin{cases} K_{AT}(N_A, K_{AB}) \\ K_{AB}(N_A, N_B) \end{cases}$
- $B \to A : K_{AB}(N_A)$

то после выполнения такого КП будет достигнута аутентификация обоих участников и полученного K_{AB}.

4.6.6 Yahalom

1. $A \to B: \ A, \ N_A$
2. $B \to T: \begin{cases} B \\ K_{BT}(A, N_A, N_B) \end{cases}$
3. $T \to A: \begin{cases} K_{AT}(B, K_{AB}, N_A, N_B) \\ K_{BT}(A, K_{AB}) \end{cases}$
4. $A \to B: \begin{cases} K_{BT}(A, K_{AB}) \\ K_{AB}(N_B) \end{cases}$

4.6.7 Ву-Лам

1. $A \to T: \ A, \ B$
2. $T \to A: S_T(K_B^+)$
3. $A \to B: K_B^+(A, N_A)$
4. $B \to T: \begin{cases} A \\ B \\ K_T^+(N_A) \end{cases}$
5. $T \to B: \begin{cases} S_T(K_A^+) \\ K_B^+(S_T(N_A, K_{AB}, A, B)) \end{cases}$
6. $B \to A: K_A^+(S_T(N_A, K_{AB}, A, B), N_B)$
7. $A \to B: K_{AB}(N_B)$

4.6.8 Needham-Schroeder с использованием асимметричной шифрсистемы

1. $A \to B: K_B^+(A, N_A)$
2. $B \to A: K_A^+(N_A, N_B)$
3. $A \to B: K_B^+(N_B)$

A и B вычисляют $K_{AB} := N_A \oplus N_B$.

4.6.9 Needham-Schroeder с использованием симметричной шифрсистемы

1. $A \to T:\ A,\ B,\ N_A$
2. $T \to A: K_{AT}\begin{pmatrix} N_A \\ B \\ K_{AB} \\ K_{BT}(K_{AB}, A, t) \end{pmatrix}$
3. $A \to B: K_{BT}(K_{AB}, A)$
4. $B \to A: K_{AB}(N_B)$
5. $A \to B: K_{AB}(N_B - 1)$

4.6.10 Ньюман-Стабблбайн

1. $A \to B:\ A,\ N_A$
2. $B \to T: \begin{cases} B \\ N_B \\ K_{BT}(A, N_A, t) \end{cases}$
3. $T \to A: \begin{cases} K_{AT}(B, N_A, K_{AB}, t) \\ K_{BT}(A, K_{AB}, t) \\ N_B \end{cases}$
4. $A \to B: \begin{cases} K_{BT}(A, K_{AB}, t) \\ K_{AB}(N_B) \end{cases}$

Данный КП обеспечивает взаимную аутентификацию. Он устойчив по отношению к атакам, основанным на десинхронизации часов (которая может произойти из-за сбоя системы или саботажа).

После того, как сеанс связи завершён, A и B могут ещё раз сделать взаимную аутентификацию (для предотвращения атаки с повторной передачей):

1. $A \to B: \begin{cases} K_{BT}(A, K_{AB}, t) \\ N'_A \end{cases}$

2. $B \to A : \begin{cases} K_{AB}(N'_A) \\ N'_B \end{cases}$

3. $A \to B : K_{AB}(N'_B)$

4.6.11 Kerberos

Базовый протокол:

1. $A \to T : \ A, \ B, \ N$

2. $T \to A : \begin{cases} K_{AT}(K_{AB}, N, L, B) & (L = \text{lifetime}) \\ K_{BT}(K_{AB}, A, L) \end{cases}$

3. $A \to B : \begin{cases} K_{BT}(K_{AB}, A, L) \\ K_{AB}(A, t, N') \end{cases}$

4. $B \to A : K_{AB}(t, N')$

Основной протокол предполагает работу с несколькими серверами:

- AS – authentication server
- $TGS_1, \ldots, TGS_n$ – tickets grant servers

1. $A \to AS : \begin{cases} A \\ TGS_i \\ N \end{cases}$

2. $AS \to A : \begin{cases} K_{A,AS}(K_{A,TGS_i}, N, L_1, TGS_i) \\ K_{AS,TGS_i}(K_{A,TGS_i}, A, L_1) \end{cases}$

3. $A \to TGS_i : \begin{cases} K_{AS,TGS_i}(K_{A,TGS_i}, A, L_1) \\ K_{A,TGS_i}(A, t_1) \end{cases}$

4. $TGS_i \to A : \begin{cases} K_{A,TGS_i}(K_{AB}, L_2, B) \\ K_{B,TGS_i}(K_{AB}, A, L_2) & (\text{билет}) \end{cases}$

5. $A \to B : \begin{cases} K_{B,TGS_i}(K_{AB}, A, L_2) \\ K_{AB}(A, t_2, N') \end{cases}$

6. $B \to A : K_{AB}(t_3, N')$

4.7 Метки времени

Иногда к передаваемым сообщениям добавляются **метки времени (МВ)**. Это делается

- для повышения доверия к аутентичности получаемых сообщений и их источников, а также
- для противодействия атакам, основанным на повторной передаче уже переданных сообщений.

МВ в сообщениях могут быть ложными или поддельными. Поэтому в некоторых КП предусмотрено проставление МВ не самим отправителем, а доверенным посредником T. Например, проставление МВ на сообщение m может делаться следующим образом.

1. $A \to T : m$
2. $T \to A : S_T(m, t)$

Однако такой способ проставления МВ не помогает в том случае, когда T по сговору с A может проставлять ложные МВ. Некоторое противодействие атакам подобного типа может оказывать нижеследующий КП, в котором проставляемые МВ связываются друг с другом. Если участнику A необходимо проставить МВ на сообщение m, то он выполняет следующие действия.

1. $A \to T : h(m), A$, где h – некоторая ХФ
2. $T \to A : S_T(A, n, h_n, t_n, A_{n-1}, h_{n-1}, t_{n-1}, H_n)$, где
   - n – порядковый номер проставляемой МВ
   - $h_n := h(m)$
   - t_n – проставленная МВ
   - A_{n-1} – участник, которому T проставил предыдущую МВ
   - h_{n-1} – значение ХФ h на сообщении участника A_{n-1}
   - t_{n-1} – предыдущая МВ
   - $H_n := h(A_{n-1}, h_{n-1}, t_{n-1}, H_{n-1})$

После того, как T проставит $n+1$–ю МВ, он выполняет действие

$$T \to A : A_{n+1}$$

где A_{n+1} – следующий после A участник, обратившийся к T за проставлением МВ.

В том случае, когда правильность МВ t_n вызывает у кого-либо сомнения, A связывается с A_{n-1} и A_{n+1} для получения информации, подтверждающей правильность t_n. Если правильность и этой информации подвергается сомнению, то A связывается с A_{n-2} и A_{n+2}, и т.д.

Можно усилить предыдущий КП, связывая текущую проставляемую МВ t_n с k предыдущими и k последующими МВ. В этом КП участник A_n хранит МВ t_{n+i}, где $i = \pm 1, \ldots, \pm k$.

В случае отсутствия доверенного посредника T для проставления МВ могут привлекаться обычные участники. В этом случае проставление МВ на сообщении m участника A может иметь следующий вид. A выбирает случайным образом некоторое множество участников $A_1, \ldots, A_k$, и посылает каждому из них строку $h(m)$ с просьбой прислать ему ЭП $S_{A_i}(h(m), t_i)$, где t_i – показания часов участника A_i в текущий момент времени. МВ сообщения m в этом случае имеет вид

$$\{S_{A_i}(h(m), t_i) \mid i = 1, \ldots, k\}$$

Глава 5

Электронная подпись

5.1 Понятие ЭП

Электронная подпись (ЭП) – это алгоритм, преобразующий каждое сообщение m в некоторую строку,

- обозначаемую знакосочетанием $S(m)$, и
- называемую **электронной подписью** сообщения m.

Алгоритм вычисления ЭП является параметрическим, в число его параметров входит **закрытый ключ** K^-, известный только подписывающему участнику.

Алгоритм ЭП должен обладать следующими свойствами.

1. Существует открытый алгоритм **проверки подлинности ЭП**, т.е. проверки для каждой пары строк m, s истинности равенства

$$S(m) = s$$

   Данный алгоритм тоже является параметрическим, в число его параметров входит **открытый ключ** K^+, и не входит K^-.

2. **ЭП невозможно подделать**, т.е. задача вычисления $S(m)$ без знания K^- является труднорешаемой.

3. Подписанное сообщение **невозможно подменить**, т.е. задача нахождения по заданной ЭП $S(m)$ такого сообщения $m' \neq m$, что $S(m') = S(m)$, является труднорешаемой.

ЭП $S(m)$ обеспечивает

- контроль целостности m, и

- аутентификацию сообщения m и его отправителя.

Иногда вместо $S(m)$ лучше вычислять $S(f(m))$, где f – ОФ. В качестве такой f обычно берут

- некоторую хэш-функцию, или
- функцию, вносящую избыточность в сообщение.

Если ЭП вычисляется участником A, то для обозначения такой ЭП может использоваться знакосочетание $S_A(m)$.

5.2 ЭП на основе асимметричных ШС

В том случае, когда подписывающий и проверяющий участники могут использовать одну и ту же асимметричную шифрсистему, $S_A(m)$ может иметь один из следующих видов:

- $K_A^-(m)$
- $K_B^+(K_A^-(m))$
- $K_A^-(A, K_A^-(m))$
- $K_T^-(A,\ m,\ K_A^-(m),\ t)$, и т. п.

Проверка подлинности: $K_A^+(s) = m$, и т. д.

Если m должно быть подписано несколькими участниками $A_1, \ldots, A_n$ одновременно, то это можно сделать

- вычислением ЭП каждого участника (недостаток такой ЭП - большой размер), или
- вычислением общей ЭП:

$$S_{A_1,\ldots,A_n}(m) = K_{A_1}^-(\ldots(K_{A_n}^-(m)))$$

(недостаток - невозможность раздельной проверки)

Если при вычислении ЭП применяется дополнительное шифрование, то рекомендуется

- выбирать такой алгоритм шифрования, который сильно отличается от алгоритма вычисления ЭП
- вычислять $K(S(m))$, а не $S(K(m))$, ибо во втором случае противник имеет пару (ОТ, ШТ).

5.3 ПЭП Фиата-Шамира

Этот ПЭП получается из ПА Фейге-Фиата-Шамира заменой случайного кортежа $(b_1, \ldots, b_k) \underset{r}{\in} \{0,1\}^k$, который генерирует V после получения x от P, на $h(m, x)$, где h – некоторая хэш-функция вида

$$h : \{0,1\}^* \to \{0,1\}^k$$

Параметры:

- p, q – простые числа, $n := pq$
  (ниже все вычисления и сравнения - в $\mathbf{Z}_n$)
- $K^- = (s_1, \ldots, s_k) \underset{r}{\in} \{2, \ldots, n-2\}^k$
- $K^+ = \left(n,\ \nu_1,\ \ldots,\ \nu_k,\ h \right)$, где

$$\forall i = 1, \ldots, k \quad \nu_i = (s_i^2)^{-1}$$

Для вычисления $S(m)$

- генерируется $z \underset{r}{\in} \{1, \ldots, n-1\}$
- вычисляется $x := z^2$
- вычисляется $b = (b_1, \ldots, b_k) := h(m, x)$

$S(m) := (b, y)$, где

$$y := z \cdot (s_1^{b_1} \cdot \ldots \cdot s_k^{b_k})$$

Проверка подлинности:

$$b = h(m, y^2 \cdot \nu_1^{b_1} \cdot \ldots \cdot \nu_k^{b_k})$$

Этот ПЭП проще чем ПЭП RSA, и работает быстрее.

5.4 ПЭП Фейге-Фиата-Шамира

Этот ПЭП получается из ПА Фейге-Фиата-Шамира заменой совокупности битовых кортежей

$$(b_{11}, \ldots, b_{1k}), \ldots, (b_{t1}, \ldots, b_{tk})$$

которые генерирует V на каждом из t раундов после получения z_i^2 от P, на битовую строку

$$(b_{11}, \ldots, b_{tk})$$

которая представляет собой последовательность первых tk битов строки $h(m, x_1, \ldots, x_t)$, где h – некоторая хэш-функция.

Параметры:

- n – число вида pq
  (ниже все вычисления и сравнения - по mod n)
- $K^- = (s_1, \ldots, s_k) \underset{r}{\in} \{2, \ldots, n-2\}^k$
- $K^+ = (n, \nu_1, \ldots, \nu_k, h)$, где

$$\forall i = 1, \ldots, k \quad \nu_i := (s_i^2)^{-1}$$

Для вычисления $S(m)$

- генерируются числа

$$z_1, \ldots, z_t \underset{r}{\in} \{1, \ldots, n-1\}$$

- вычисляются числа $x_1, \ldots, x_t$, где

$$\forall i = 1, \ldots, t \quad x_i = z_i^2$$

- вычисляются биты $(b_{11}, \ldots, b_{tk})$ = первые tk битов в $h(m, x_1, \ldots, x_t)$

$S(m) := \begin{pmatrix} y_1, \ldots, y_t \\ b_{11}, \ldots, b_{tk} \end{pmatrix}$, где $\forall i = 1, \ldots, t$

$$y_i := z_i \cdot (s_1^{b_{i1}} \cdot \ldots \cdot s_k^{b_{ik}})$$

Проверка подлинности:

$$(b_{11}, \ldots, b_{tk}) = \text{ первые } tk \text{ битов в } h(m, u_1, \ldots, u_t)$$

где $\forall i = 1, \ldots, t \quad u_i = y_i^2 \cdot (\nu_1^{b_{i1}} \cdot \ldots \cdot \nu_k^{b_{ik}})$.

Вероятность обмана не превосходит 2^{-kt}, она зависит от сложности разложения n на множители.

Рекомендуется брать $k = 9, t = 8$, и $\nu_1, \ldots, \nu_k$ = первые k простых чисел.

На этот ПЭП м.б. атаки, основанные на парадоксе дней рождения.

5.5 ПЭП Гиллу-Кискате

Этот ПЭП получается из ПА Гиллу-Кискате заменой случайного значения $b \underset{r}{\in} \{0, \ldots, k-1\}$, которое генерирует V после получения $x := z^k$ от P, на $h(m, x)$, где h – некоторая хэш-функция вида

$$h : \mathbf{Z} \to \mathbf{Z}_k$$

Параметры:

- p, q – простые числа, $n := pq$
  (ниже все вычисления и сравнения - по mod n)
- $K^- = s \underset{r}{\in} \{2, \ldots, n-2\}$
- $K^+ = (\ n,\ k,\ \nu := (s^k)^{-1}\ )$

$$S(m) := \begin{cases} b := h(m, z^k), & \text{где } z \underset{r}{\in} \{1, \ldots, n-1\} \\ y := z \cdot s^b \end{cases}$$

Проверка подлинности:

$$b = h(m, y^k \cdot \nu^b)$$

Вместо одного z м.б. $\{z_i \mid i = 1, \ldots, t\}$.

5.6 ПЭП Эль-Гамаля

Параметры:

- p – простое число
- $K^- = a \underset{r}{\in} \{1, \ldots, p-2\}$
- $K^+ = (p, \alpha, \beta)$, где
  - $\alpha \in \mathbf{Z}_p$ – примитивный элемент
  - $\beta := \alpha^a \ \% \ p$.

$S(m) := (\gamma, \delta)$, где:

- $\gamma := \alpha^i \ \% \ p$, где $i \underset{r}{\in} \mathbf{Z}^*_{p-1}$
- δ ищется из уравнения

$$m \underset{p-1}{=} a\gamma + i\delta$$

Проверка подлинности:

$$\beta^\gamma \cdot \gamma^\delta \underset{p}{=} \alpha^m$$

Можно уменьшить длину ЭП, заменив (γ, δ) на

$$(\gamma \ \% \ q,\ \delta \ \% \ q)$$

где q – простое, и $q \mid p-1$. В этом случае условие проверки подлинности имеет вид

$$\alpha^m \cdot \beta^{-\gamma} \ \% \ p \underset{q}{=} \gamma^\delta$$

5.7 ПЭП Шнорра

Параметры:

- p, q – простые числа, где
  - $|p| \sim 512$ бит,
  - $|q| \sim 140$ бит, и
  - $q|(p-1)$
- $K^- = x \underset{r}{\in} \mathbf{Z}_q^*$
- $K^+ = (p, q, g, y, h)$, где
  - $g \in \mathbf{Z}_p \setminus \{1\}, \quad g^q \underset{p}{=} 1$
  - $y := g^{-x} \pmod p$
  - h – хэш-функция вида

$$h : \{0,1\}^* \to \{0,1\}^t$$

$S(m) := (e, s)$, где

$$\begin{cases} e = h(g^k \ \% \ p, m), \text{ где } k \underset{r}{\in} \mathbf{Z}_q^* \\ s := (k + xe) \ \% \ q \end{cases}$$

Проверка подлинности:

$$e = h(g^s y^e \ \% \ p, m)$$

Сравнение ПЭП Шнорра с другими ПЭП:

- уровень безопасности этого ПЭП – такой же, как у ПЭП RSA и Эль-Гамаля
- длина ЭП Шнорра меньше, чем длина ЭП RSA и Эль-Гамаля (более точно, |ЭПШ| = 212 бит, $\sim 1/2$|ЭП RSA|).

5.8 ПЭП DSA

5.8.1 Основной алгоритм DSA

DSA = Digital Signature Algorithm.

Параметры:

- p, q – простые числа, где
  - $|p| = L \in \{512, \ldots, 1024\}$, L делится на 64
  - $|q|$ = примерно 160
  - $q|(p-1)$
- $K^- = x < q$ (160 бит)
- $K^+ = (p, q, g, y, h)$, где
  - $g = \eta^{(p-1)/q} \ \% \ p$, где
    - $\eta \underset{r}{\in} \{1, \ldots, p-2\}$,
    - $g > 1$
  - $y := g^x \ \% \ p$
  - h – хэш-функция

$S(m) := (r, s)$, где

$$\begin{cases} r := (g^k \ \% \ p) \ \% \ q, \text{ где } k \underset{r}{\in} \mathbf{Z}_q^* \\ s := (k^{-1} \cdot (h(m) + xr)) \ \% \ q \end{cases}$$

(Можно заранее создать массив случайных значений k, и вычислить соответствующие r и k^{-1}.)

Проверка подлинности:

$$r = v$$

где

$$\left.\begin{array}{l} v := ((g^{u_1} \cdot y^{u_2}) \ \% \ p) \ \% \ q \\ u_1 := (h(m) \cdot w) \ \% \ q \\ u_2 := (r \cdot w) \ \% \ q \\ w := s^{-1} \pmod{q} \end{array}\right\} \tag{5.1}$$

5.8.2 Варианты DSA

1. $S(m) := (r, s, t)$, где

$$\begin{cases} s := (h(m) + xr) \cdot d \ \% \ q \\ t := k \cdot d \ \% \ q \end{cases}$$

где $k, d \underset{r}{\in} \{1, \ldots, q-1\}$.

Проверка подлинности:

$$r = v$$

где

- v, u_1, u_2 – как в (5.1)
- $w = t \cdot s^{-1} \pmod q$

2. $S(m) := (r, s)$, где

$$\begin{cases} s := k \cdot (h(m) + xr)^{-1} \pmod q \\ k \underset{r}{\in} \mathbf{Z}_q^* \end{cases}$$

Проверка подлинности:

$$r = v$$

где

- $u_1 := (h(m) \cdot s) \ \% \ q$
- $u_2 := s \cdot r \ \% \ q$

5.8.3 Генерация простых p, q для DSA

$L - 1 =: 160n + b$, где $0 \leq b < 160$.

1. $s \underset{r}{\in} \{0,1\}^{\geq 160}, \ g := |s|$
2. $U := SHA(s) \oplus SHA((s+1) \ \% \ 2^g)$
   (алгоритм SHA содержит скрытый канал, позволяющий встроить в ЭП секретное сообщение)
3. $q := U$[1-й и последний биты := 1]
4. (q не простое)? go to (1)
5. $c := 0, \ N := 2$
6. $\forall k = 0, \ldots, n \quad V_k := SHA((s + N + k) \ \% \ 2^g)$
7. $$\begin{aligned} W &:= V_0 + \\ &+ 2^{160} \cdot V_1 + \ldots + \\ &+ 2^{160(n-1)} \cdot V_{n-1} + \\ &+ 2^{160n} \cdot (V_n \ \% \ 2^b) \end{aligned}$$

8. $X := W + 2^{L-1}$ (L бит)
9. $p := X - ((X \ \% \ 2q) - 1) \quad (\underset{2q}{=} 1)$
10. ($p \geq 2^{L-1}$ и p – простое) ? return (p, q, s, c)
    (зная s и c можно восстановить процедуру генерации p и q)
11. $c := c + 1, \ N := N + n - 1$
12. ($c = 2^{12}$) ? go to (1) : go to (6)

5.8.4 Вариант алгоритма генерации простых p, q для DSA

1-4. – как выше.

5. $p :=$ конкатенация $(q, s, C, SHA(s))$, где
$$C = (\underbrace{0 \ldots 0}_{32 \text{ шт.}})$$
6. $p := p - (p \ \% \ q) + 1$
7. $p := p + q$
8. $\Big($(часть C в p) $= 0x7\underbrace{f \ldots f}_{7 \text{ шт.}}\Big)$? go to (1)
9. (p – не простое) ? go to (7)

Преимущества: не надо хранить s и C (ибо они $\subseteq p$). Это хорошо для устройств с небольшой памятью (smart cards).

5.8.5 Реализация некоторых криптографических преобразований на базе реализации DSA

Если алгоритм DSA реализован каким-либо образом (например, в виде спецпроцессора), то эту реализацию можно использовать для вычисления некоторых криптографических преобразований. Мы покажем, как можно использовать реализацию DSA для

- шифрования и дешифрования в системе RSA
- шифрования и дешифрования в системе Эль-Гамаля.

Напомним, что алгоритм DSA оперирует со следующими параметрами:

$$p, \ q, \ x, \ g, \ \eta, \ k, \ h$$

где $p, \ q, \ x, \ g, \ \eta, \ k$ – числа, а h – хэш-функция.

Реализация шифрования и дешифрования RSA на базе реализации DSA

Если требуется реализовать шифрование и дешифрование в системе RSA с модулем n, ОК e и ЗК d, то

- шифрование ОТ m осуществляется следующим образом:
  - $(r, s) := DSA \begin{bmatrix} p := n \\ q := n \\ g := m \\ k := e \\ x := 0 \\ \eta := 0 \end{bmatrix}$
  - ШТ $:= r$
- дешифрование ШТ m осуществляется следующим образом:
  - $(r, s) := DSA \begin{bmatrix} p := n \\ q := n \\ g := m \\ k := d \\ x := 0 \\ \eta := 0 \end{bmatrix}$
  - ОТ $:= r$

Реализация шифрования и дешифрования Эль-Гамаля на базе реализации DSA

Если требуется реализовать шифрование и дешифрование в системе Эль-Гамаля с модулем p, ЗК x и ОК y, то

- шифрование ОТ m осуществляется следующим образом:
  - $(r, s) := DSA \begin{bmatrix} q := p \\ x := 0 \\ \eta := 0 \end{bmatrix}, \quad a := r$

- $(r, s) := DSA \begin{bmatrix} q := p \\ g := y \\ x := 0 \\ \eta := 0 \end{bmatrix}, \quad u := r$
- $(r, s) := DSA \begin{bmatrix} q := p \\ g := m \\ k := 1 \\ x := u \\ \eta := 0 \end{bmatrix}, \quad b := s$
- ШТ $:= (a, b)$

- дешифрование ШТ (a, b) осуществляется следующим образом:
  - $(r, s) := DSA \begin{bmatrix} q := p \\ g := a \\ k := x \\ x := 0 \\ \eta := 0 \end{bmatrix}, \quad e := r$
  - $(r, s) := DSA \begin{bmatrix} q := p \\ g := 1 \\ k := e \\ x := b \\ \eta := 0 \end{bmatrix}$
  - ОТ $:= s$

5.9 ПЭП ГОСТ (старый)

Параметры - те же, что и у DSA, кроме η, причём

- $|q| = 256$ бит,
- $g < p - 1,\ g^q \underset{p}{=} 1$

$S(m) := (r, s)$, где

$$\begin{cases} r := (g^k \ \% \ p) \ \% \ q \ \% \ 2^{256} \\ s := (k \cdot h(m) + xr) \ \% \ q \ \% \ 2^{256} \end{cases}$$

где $k \underset{r}{\in} \{1, \ldots, q-1\}$, и

- если $h(m) \underset{q}{=} 0$, то полагаем $h(m) := 1$, и
- если получилось так, что $r = 0$ или $s = 0$, то пробуем другое k.

Проверка подлинности:

$$((g^{z_1} \cdot y^{z_2}) \ \% \ p) \underset{q}{=} h(m)^{q-2}$$

где

$$\begin{cases} z_1 := (s \cdot v) \% q \\ z_2 := ((q - r) \cdot v) \% q \end{cases}$$

Новый ПЭП ГОСТ основан на эллиптических кривых.

5.10 ПЭП ESIGN

Параметры:

- p, q – простые числа длины ≥ 192 бит
- $K^- = (p, q)$
- $K^+ = \left(\ n := p^2 q, \ k := 2^i, \ h \ \right)$, где
  - $i \in \{3, \ldots, 10\}$, и
  - $h : \mathbf{Z} \to \mathbf{Z}_n$ – хэш-функция.

$S(m) := s$, где

$$s := x + \left(\left(\left[\frac{h(m) - u}{pq} \right] \cdot v \right) \ \% \ p \right) \cdot pq$$

где

- $x \underset{r}{\in} \{1, \ldots, pq - 1\}$
- $u := x^k \ \% \ n$
- $v := \dfrac{1}{k \cdot x^{k-1}} \ \% \ p$

Проверка подлинности:

$$h(m) \ \leq \ s^k \ \% \ n \ < \ h(m) + 2^a$$

где

$$a = \left[\frac{2 \cdot \text{число битов } n}{3} \right]$$

Для ускорения вычисления $S(m)$ можно заранее

- сгенерировать массив значений x, и
- вычислить соответствующие значения u и v.

Сравнение с ПЭП RSA, DSA, El Gamal, Рабина:

- скорость – гораздо больше
- стойкость – не меньше
- размер ключа – примерно такой же
- размер ЭП – примерно такой же

5.11 Undeniable ЭП

Иногда участник A, подписывающий сообщения, не хочет, чтобы любой другой участник имел возможность самостоятельно убедиться в подлинности его ЭП. Он хочет, чтобы

- только сам A имел возможность доказать любому другому участнику B подлинность своей ЭП, и
- участник B не мог бы использовать это доказательство для того, чтобы доказать любому другому участнику подлинность этой ЭП.

ЭП такого вида называется **undeniable**.

5.11.1 Протокол Шаума undeniable ЭП

Один из возможных протоколов undeniable ЭП был придуман Шаумом (Chaum).

Параметры:

- p – большое простое число
  (ниже все вычисления и сравнения - в $\mathbf{Z}_p$)
- $K^- := z$

- $K^+ := (p, g, \eta)$, где
  - $g \in \mathbf{Z}_p^*$ – примитивный элемент (т.е. $\langle g \rangle = \mathbf{Z}_p^*$), и
  - $\eta := g^z$

Предполагается, что подписываемое сообщение m является элементом $\mathbf{Z}_p$ (если m слишком велико, то подписывать можно не m а $h(m)$).

$$S(m) := s := m^z$$

Проверка подлинности:

- $B \to A : c := s^u \eta^v$, где $u, v \underset{r}{\in} \mathbf{Z}_p^*$
- $A \to B : d := c^t$, где $t := z^{-1} \ \% \ (p-1)$
- B принимает ЭП iff
$$m^u g^v = d$$

Если B попытается убедить другого участника в подлиннности ЭП $S(m)$ путём показа записи исполнения этого КП, то это ему не удастся: он мог её и подделать.

5.11.2 Отрицание undeniable ЭП

Иногда для участника A, который подписывает сообщения по протоколу, описанному в предыдущей секции, представляет интерес возможность отрицать утверждение о том, что некоторая строка s якобы является ЭП сообщения m, которая была вычислена им по этому ПЭП, т.е. его интересует возможность доказывать утверждение

$$s \neq S(m) \tag{5.2}$$

Данное утверждение может быть доказано, например, следующим образом (ниже используются обозначения и предположения из предыдущей секции).

- $B \to A : c := m^u g^v$, где $u, v \underset{r}{\in} \mathbf{Z}_p^*$
- $A \to B : \begin{cases} s_1 := cg^q \\ s_2 := (cg^q)^x \end{cases}$, где $q \underset{r}{\in} \mathbf{Z}_p^*$
- $B \to A : u, v$

- $A \to B : q$
- B принимает доказательство утверждения (5.2) iff

$$\left\{ \begin{array}{l} s_1 = c \cdot g^q \\ s_2 = (g^x)^{v+q} \cdot s^u \end{array} \right\}$$

Данное доказательство является Д0Р.

5.11.3 Другой протокол undeniable ЭП

Параметры:

- p – простое число
  (ниже все операции и сравнения - по $\bmod p$)
- q – простое число, такое, что $q|(p-1)$
- $g := \eta^{(p-1)/q}$, где $\eta \underset{r}{\in} \mathbf{Z}_p^*$
  если $\neg(1 < g < q)$, то выбирается другое η
- $K^- = (x, z)$, где $x \neq z \in \mathbf{Z}_q$
- $K^+ = (\ p,\ q,\ g,\ y := g^x,\ u := g^z\ )$.

$$S(m) := (\ r,\ s,\ v\ )$$

где

- $r := g^i,\ v := g^j,\ i \underset{r}{\in} \mathbf{Z}_{p-1}^*,\ j \underset{r}{\in} \mathbf{Z}_q^*$
- s вычисляется как решение сравнения

$$si + rx \underset{q}{=} vjzm$$

Проверка подлинности:

- $B \to A : c := (v^{vma} \cdot g^b)$, где $a, b \underset{r}{\in} \mathbf{Z}_p^*$
- $A \to B : \left\{ \begin{array}{ll} h_1 := cg^k, & \text{где } k \underset{r}{\in} \mathbf{Z}_p^* \\ h_2 := h_1^z & \end{array} \right.$
- $B \to A : a, b$
- $A \to B : (c = (v^{vma} \cdot g^b))\ ?\ k$

- B принимает ЭП iff
  - $h_1 = v^{vma} \cdot g^{b+k}$, и
  - $h_2 = y^{ra} \cdot r^{sa} \cdot u^{b+k}$

Эту ЭП тоже можно отрицать.
Если A опубликует z, то будет обычный ПЭП.

5.12 ЭП, подтверждаемая уполномоченным участником

Ещё один вид ограничений на возможность проверки ЭП заключается в следующем. Участник A, подписывающий сообщения, хочет, чтобы подлинность его ЭП могли доказывать только

- он сам, а также
- специальный участник C, называемый **уполномоченным** участником (**designated confimer**)

5.12.1 Вычисление ЭП

Параметры:

- p – большое простое число
- $K_C^- := z \underset{r}{\in} \mathbf{Z}_p^*$
- $K_A^- := (p_1, p_2)$, где p_1, p_2 – простые числа
- $K^+ = (p, g, \eta, n, h)$, где
  - $g \in \mathbf{Z}_p^*$ – примитивный элемент
  - $\eta := g^z \ \% \ p$
  - $n := p_1 \cdot p_2$,
  - h – хэш-функция

$S(m) := (a, b, j)$ где

- $a := g^x \ \% \ p$, где $x \underset{r}{\in} \mathbf{Z}_p^*$
- $b := \eta^x \ \% \ p$
- $j := (h(m) \oplus h(a, b))^{1/3} \pmod n$
  (поскольку A знает разложение n на простые множители, то он может быстро вычислять функцию $x^{1/3}$ в $\mathbf{Z}_n$)

5.12.2 Доказательство подлинности ЭП участником A

- $B \to A : c := g^u \eta^v \ \% \ p$, где $u, v \underset{r}{\in} \mathbf{Z}_p^*$
- $A \to B : \begin{cases} d := g^q \ \% \ p, & \text{где } q \underset{r}{\in} \mathbf{Z}_p^* \\ e := (cd)^x \ \% \ p \end{cases}$
- $B \to A : u, v$
- $A \to B : (g^u \eta^v \underset{p}{=} c) \ ? \ q$
- B принимает ЭП iff
  - $g^q \underset{p}{=} d$
  - $e / a^q \underset{p}{=} a^u b^v$
  - $h(m) \oplus h(a, b) \underset{n}{=} j^3$

Если B попытается убедить другого участника в подлиннности ЭП $S(m)$ путём показа записи исполнения этого КП, то это ему не удастся: он мог её и подделать.

5.12.3 Доказательство подлинности ЭП уполномоченным участником C

C может убедить произвольного участника D в подлинности ЭП $S(m) := (a, b, j)$ следующим образом:

- $D \to C : c := g^u a^v \ \% \ p$, где $u, v \underset{r}{\in} \mathbf{Z}_p$
- $C \to D : \begin{cases} d := g^q \ \% \ p, & \text{где } q \underset{r}{\in} \mathbf{Z}_p^* \\ e := (cd)^z \ \% \ p \end{cases}$
- $D \to C : u, v$
- $C \to D : (g^u a^v \underset{p}{=} c) \ ? \ q$
- D принимает ЭП iff
  - $g^q \underset{p}{=} d$
  - $e / \eta^q \underset{p}{=} \eta^u b^v$

– $h(m) \oplus h(a,b) \underset{n}{=} j^3$

На базе этого ПЭП и некоторой схемы разделения секрета можно построить такой ПЭП, в котором доказывать подлинность ЭП A могут любые m участников из $\{C_1, \ldots, C_n\}$).

5.13 Протоколы ЭП Лампорта

5.13.1 Одноразовый ПЭП Лампорта

Предполагается, что каждое подписываемое сообщение имеет длину r.

Параметры:

- $K^- = \{x_{ij} \underset{r}{\in} \{0,1\}^n \mid i = 1, \ldots, r;\ j = 0,1\}$
- $K^+ = \begin{cases} \text{ОФ}\ \ f : \{0,1\}^n \to \{0,1\}^n \\ \{y_{ij} := f(x_{ij}) \mid i = 1, \ldots, r;\ j = 0,1\} \end{cases}$

Функция вычисления ЭП имеет вид

$$S : \{0,1\}^r \to \{0,1\}^{n \cdot r}$$

и определяется следующим образом: для каждого сообщения $m = (m_1, \ldots, m_r) \in \{0,1\}^r$

$$S(m) := (s_1, \ldots, s_r) := (x_{1m_1}, \ldots, x_{rm_r})$$

Проверка подлинности:

$$(f(s_1), \ldots, f(s_r)) \ = \ (y_{1m_1}, \ldots, y_{rm_r})$$

Недостатки:

- этот ПЭП одноразовый, после подписи каждого сообщения надо генерировать новые ключи
- размер подписи в несколько раз больше размера подписываемого сообщения:

  $$|S(m)| = n \cdot |m|$$

  (должно быть наоборот)

Достоинства: стойкость. Если противник умеет подделывать такую ЭП за полиномиальное время, то он умеет инвертировать f за полиномиальное время.

5.13.2 Многоразовый ПЭП Лампорта

Подписываемые сообщения $= m_1, m_2, \ldots$, где

$$\forall i \geq 1 \quad m_i \in \{0,1\}^r$$

Для подписи каждого m_i генерируются свои K_i^- и K_i^+, которые имеют тот же вид, что и в предыдущем ПЭП.

K_i^- делится на 2 части: K_{iL}^- и K_{iR}^-, в пропорции $(r : n-r)$.

$$P \to V : \begin{pmatrix} m_i \\ K_{i+1}^+ \\ S_{K_{iL}^-}(m_i) \\ S_{K_{iR}^-}(h(Y_{i+1}) \end{pmatrix}$$

где h – случайно выбранная хэш-функция вида

$$h : \{0,1\}^{2nr} \to \{0,1\}^{n-r}$$

5.13.3 ПЭП Диффи-Лампорта

Этот ПЭП тоже одноразовый, т.к. после проверки V узнаёт часть K^-.

В этом ПЭП используется некоторая симметричная система шифрования.

Параметры:

- $n =$ длина подписываемых сообщений
- $K^- = \{K_{ij} \mid i = 1, \ldots, n, \ j = 0,1\}$ – набор ключей симметричной системы шифрования

  (можно использовать один ЗК K, и считать, что $K_{ij} = K(i,j)$)
- $K^+ = (\sigma, \rho)$, где
  - $\sigma = \{\sigma_{ij} \underset{r}{\in} \{0,1\}^n \mid i = 1, \ldots, n, \ j = 0,1\}$
  - $\rho = \{\rho_{ij} := K_{ij}(\sigma_{ij}) \mid i = 1, \ldots, n; \ j = 0,1\}$

Если подписываемое сообщение m имеет вид $(m_1, \ldots, m_n)$, то

$$S(m) := (K_{1m_1}, \ldots, K_{nm_n})$$

Проверка подлинности:

$$\forall i = 1, \ldots, n \quad \rho_{im_i} = K_{im_i}(\sigma_{im_i})$$

Недостаток: размер подписи больше, чем размер подписываемого сообщения. Для устранения этого недостатка можно подписывать не m, а $h(m)$.

5.14 ПЭП Онга-Шнорра-Шамира

Параметры:

- n – большое число
  (ниже все операции и сравнения - по $\bmod n$)
- $K^- = k \underset{r}{\in} \mathbf{Z}_n^*$
- $K^+ = (n, k^2)$

$S(m) := (s_1, s_2)$, где

$$\begin{cases} s_1 := \frac{1}{2} \cdot \left(\frac{m}{i} + i\right) & (i \underset{r}{\in} \mathbf{Z}_n^*) \\ s_2 := \frac{k}{2} \cdot \left(\frac{m}{i} - i\right) \end{cases}$$

Проверка подлинности:

$$m = s_1^2 - \frac{s_2^2}{k^2}$$

Этот ПЭП нестоек.

5.15 Слепая (blinded) ЭП

A хочет, чтобы B, не зная m, подписал его.

Пример (Chaum):

Выбирается открытое число $n = pq$. Ниже все числа и вычисления с ними - в $\mathbf{Z}_n$. $K_B^+ = e, K_B^- = d$ (со свойствами, как в RSA), $S_B(m) := K_B^-(m) = m^d \ \% \ n$.

- $A \to B : t := m \cdot k^e$, где $k \underset{r}{\in} \mathbf{Z}_n^*$ (маскирует)
- $B \to A : t^d \quad (= (m \cdot k^e)^d = m^d \cdot k^{ed} = m^d \cdot k)$
- A снимает маскировку $s := t^d \cdot k^{-1} \quad (= m^d)$

Пример использования СЭП:

- $A \to B : \xi_1(d_1), \ldots, \xi_n(d_n)$.
- B демаскирует (т.е. удаляет ξ_i) все сообщения кроме одного, и подписывает оставшееся сообщение.

5.16 Совместные ЭП

Иногда требуется, чтобы сообщение было подписано совместно несколькими участниками $A_1, \ldots, A_k$.

Один из возможных ПЭП, решающих эту задачу, мы рассматривали в секции 5.2. Другой протокол совместной ЭП имеет следующий вид.

Параметры:

- $K^+ = (n, v, h)$, где
  - n, v – числа (ниже все вычисления – в $\mathbf{Z}_n$)
  - $h : \mathbf{Z} \to \mathbf{Z}_v$ – хэш-функция
- $K_i^- = x_i$
- $K_i^+ = y_i := (x_i^v)^{-1}$

$$S(m) \;=\; \left(\begin{array}{l} d := h(m, T_1 \ldots T_k) \\ D := D_1 \ldots D_k \end{array} \right)$$

где $\forall i = 1, \ldots, k$

- $T_i := r_i^v \quad (r_i \underset{r}{\in} \mathbf{Z}_n \setminus \{0\})$
- $D_i := r_i x_i^d$.

Проверка подлинности:

$$d = h(m, D^v (y_1 \ldots y_k)^d)$$

5.17 Общий ПЭП

Его частные случаи – ПЭП DSA, Эль-Гамаля, Шнорра.

Параметры:

- p, q – большие простые числа, причём $q | (p-1)$
- $K^- = x < q$
- $K^+ = (g, y)$, где
  - $g \in \mathbf{Z}_p^* : g^q \underset{p}{=} 1$
  - $y := g^x \;\%\; q$

Для вычисления $S(m)$

- вычисляются
  - $r := g^k \ \% \ p$ (где $k \underset{r}{\in} \mathbf{Z}_q^*$), и
  - $r' := r \ \% \ q$
- составляется уравнение

$$ak \underset{q}{=} b + cx$$

где

$$\{a, b, c\} = \begin{bmatrix} \{\pm r', \pm s, m\} \\ \{\pm r'm, \pm s, 1\} \\ \{\pm r'm, \pm ms, 1\} \\ \{\pm mr', \pm r's, 1\} \\ \{\pm ms, \pm r's, 1\} \end{bmatrix}$$

(м.б. любой из этих вариантов, любая из 6 перестановок в каждом варианте, и любой выбор знаков).

Единственной переменной в этом уравнении является s. Данное уравнение решается, и

$$S(m) := (r, a, b, c)$$

Проверка подлинности:

$$r^a \underset{p}{=} g^b \cdot y^c$$

5.18 Встраивание скрытых сообщений в ЭП

Если обмен сообщениями между A и B контролируется таким противником W, который

- может читать все пересылаемые сообщения,
- знает все ЗК, которыми пользуются A и B, и
- запрещает участникам A и B обмениваться такими сообщениями, с содержанием которых он не сможет ознакомиться

представляет интерес задача организации такого обмена сообщениями между A и B, при котором пересылаемые сообщения могли бы передавать информацию, явно не выраженную в этих сообщениях.

Данную задачу называют задачей создания **скрытого канала (СК)**.

Например, в качестве скрытого канала может выступить функция, сопоставляющая пересылаемому сообщению какие-либо структурные характеристики этого сообщения.

В этом пункте мы рассматриваем создание скрытых каналов в ЭП.

5.18.1 СК в ПЭП Онга-Шнорра-Шамира

ПЭП Онга-Шнорра-Шамира описан в пункте 5.14.

Если B знает ЗК k, при помощи которого A создаёт ЭП для своих сообщений, то A может передать скрытое сообщение $\mu \in \mathbf{Z}_n^*$ в ЭП $S_A(m)$, где $m \in \mathbf{Z}_n^*$, полагая число i в этом ПЭП равным μ.

B может вычислить μ по формуле

$$\mu := \frac{m}{s_1 + s_2 k^{-1}}$$

5.18.2 СК в ПЭП Эль-Гамаля

ПЭП Эль-Гамаля описан в пункте 5.6.

Если B знает ЗК a, при помощи которого A создаёт ЭП для своих сообщений, то A может передать скрытое сообщение $\mu \in \mathbf{Z}_{p-1}^*$ в ЭП $S_A(m)$, где $m \in \mathbf{Z}_p^*$ и $(m, \mu) = 1$, полагая число i в этом ПЭП равным μ.

B может вычислить μ из уравнения

$$m \underset{p-1}{=} a\gamma + \mu\delta$$

5.18.3 СК в ПЭП NEW-ESIGN

ПЭП NEW-ESIGN отличается от описанного в пункте 5.10 ПЭП ESIGN тем, что простое число q в нём заменяется на пару простых чисел q_1, q_2, а в качестве n берётся произведение $p^2 q_1 q_2$.

Если B знает q_2, то A может передать скрытое сообщение $\mu \in \mathbf{Z}_{q_2}$ в ЭП $S_A(m)$, если при вычислении $S_A(m)$ в качестве x будет взято число вида $\mu + i \cdot q_2$. B может вычислить $\mu := s \ \% \ q_2$

5.18.4 СК в ПЭП DSA

ПЭП DSA описан в пункте 5.8. Напомним, что ЭП DSA имеет вид (r, s), где

$$r := (g^k \ \% \ p) \ \% \ q, \text{ где } k \underset{r}{\in} \mathbf{Z}_q^*$$

A может передать B 1 бит скрытой информации за счёт подбора такого $k \in \mathbf{Z}_q^*$, чтобы число r в $S_A(m)$ было квадратичным вычетом (или невычетом) по заданному простому числу p_1. Например, A и B могут заранее договориться, что

$$\mu = \begin{cases} 1, & \text{если } r \text{ – квадратичный вычет в } \mathbf{Z}_{p_1} \\ 0, & \text{иначе} \end{cases}$$

(отметим, что при случайно выбранном $k \in \mathbf{Z}_q^*$ вероятность того, что r – квадратичный вычет в $\mathbf{Z}_{p_1}$, примерно равна $1/2$).

Аналогичным способом можно передать в одной ЭП n битов, если подобрать k так, чтобы r представлял собой нужные квадратичные вычеты или невычеты по нескольким модулям $p_1, \ldots, p_n$.

5.19 Соглашение об обозначениях

Везде в других главах знакосочетанне вида $S(m)$ (или $S_A(m)$) будет обозначать не просто ЭП сообщения m, а пару, состоящую из

- подписываемого сообщения (т.е. m), и
- его ЭП.

Глава 6

Генерация и передача ключей

6.1 Понятие сеансового ключа

Ключи шифрования делятся на два класса: долговременные и сеансовые.

1. Долговременные ключи как правило относятся к асимметричным системам шифрования, их сложнее взломать, хотя скорость шифрования в таких ШС примерно в 1000 раз меньше чем скорость шифрования в симметричных ШС. Как правило, долговременные ключи используются для шифрования сеансовых ключей.

2. Сеансовые ключи, как правило, относятся к симметричным ШС. Скорость работы таких ШС велика, но сеансовые ключи проще взломать.

   Сеансовый ключ (СК):

   - используется в одном сеансе связи,
   - уничтожается после завершения сеанса.

   СК может генерироваться следующими способами:

   - доверенным посредником,
   - одним из участников
   - группой участников, путём их взаимодействия.

   Передача СК может производиться

   - как с аутентификацией участников, которым передаётся этот СК,
   - так и без такой аутентификации.

6.2 КП генерации сеансового ключа доверенным посредником

- $A \to T : A, B$
- $T \to A : K_{AT}(K), K_{BT}(K)$
- $A \to B : K_{BT}(K)$

Во втором и третьем сообщениях можно добавить $K_{BT}(A)$.

6.3 КП передачи сеансового ключа без доверенного посредника

КП Шамира (имеет уязвимость).

Если K_A^+ и K_B^+ коммутируют как функции, т.е.

$$\forall m \quad K_A^+(K_B^+(m)) = K_B^+(K_A^+(m)) \tag{6.1}$$

то можно использовать следующий КП передачи сеансового ключа:

- $A \to B : K_A^+(K)$
- $B \to A : K_B^+(K_A^+(K))$
- $A \to B : K_B^+(K)$

В качестве ШС можно взять RSA, но нельзя брать $K_A^+(m) := m \oplus \Gamma$.

6.4 Обновление сеансового ключа

K – старый, а K' – новый сеансовый ключ.

Без аутентификации:

1. $A \to B : K(K', t, B)$
2. $A \to B : K' \oplus h_K(t, B)$
3. $A \to B : K_B^+(K', t, A)$

С односторонней аутентификацией:

1. - $B \to A : N$
   - $A \to B : K(K', N, B)$

2.
   - $A \to B : A, N_A$
   - $B \to A : K(N_A, K')$
   - $A \to B : K'(N_A)$

С двусторонней аутентификацией:

1. пусть A и B имеют k_A и $k_B \underset{r}{\in} \ldots$
   - $B \to A : N_B$
   - $A \to B : K(k_A, N_A, N_B, B)$
   - $B \to A : K(k_B, N_A, N_B, A)$
   - A и B вычисляют $K' = f(k_A, k_B)$
2. Andrew Secure RPC Handshake
   - $A \to B : A, K(N_A)$
   - $B \to A : K(N_A + 1, N_B)$
   - $A \to B : K(N_B + 1)$
   - $B \to A : K(K', N'_B)$

   Дырка: противник M может повторить старое сообщение как последнее сообщение в КП, и убедить A использовать старый ключ K (который м.б. раскрытым).

   Для устранения уязвимости надо в последнем действии написать $K(K', N'_B, N_A)$.

6.5 Проверка одинаковости ключа

A и B имеют общие ключ K и пароль P. Они хотят проверить одинаковость ключа K. В излагаемом ниже КП используется функция h' от двух аргументов, определяемая следующим образом:

$$h'(x, y) = h(h(x, y) \,\%\, 2^m, y)$$

где запятая, разделяющая аргументы функции h, обозначает операцию конкатенации (h' имеет много коллизий по x, и не имеет коллизий по y).

КП проверки одинаковости ключа имеет следующий вид:

1. $A \to B : h'(P, K)$
2. B вычисляет своё $h'(P, K)$, и сравнивает его со значением, полученным от A (если они не совпадают, то у A и B ключи разные)

3. $B \to A : \eta := h'(h(P), K)$

4. A вычисляет $h'(h(P), K)$ и сравнивает его с η: если они не совпадают, то у A и B ключи разные.

Если на шаге 2 противник, зная $h'(P, K_A)$, попытается подобрать P, такой, что значение $h'(P, K_B)$ совпадёт с тем, что вычислит B, то он с большой вероятностью подберёт неправильный пароль (т.к. h' имеет много коллизий по x).

6.6 Распределение ключа среди части участников

Имеется участник A, который хочет передать ключ K участникам $B_1, \ldots, B_n$ посредством единой широковещательной рассылки, причём этот ключ должны получить только те из них, которые входят в заданное подмножество $\{B_{i_1}, \ldots, B_{i_k}\}$. Предполагается, что каждый из участников B_i имеет ключ K_i симметричной ШС для связи с A.

Данную задачу можно решить одним из следующих способов.

1. Если для любых различных $i, j \in \{1, \ldots, n\}$ числа K_i и K_j взаимно просты, то по китайской теореме об остатках существует число R, обладающее свойством

$$R \,\%\, K_i \;=\; (i \in \{i_1, \ldots, i_k\}) \;?\; K \;:\; 0$$

   Искомый КП имеет вид:

$$\forall i = 1, \ldots, n \quad A \to B_i : R$$

2. A создаёт пороговую СРС типа $(k + j + 1, 2k + j + 1)$, где

   - j – некоторое небольшое число
   - секретом является передаваемый ключ K, и
   - долями являются
     - ключи $K_i \quad (i \in \{i_1, \ldots, i_k\})$, а также
     - j случайно выбранных значений

   A посылает всем $k + j$ случайно выбранных долей.

   Каждый B_i добавляет свою долю к полученным, и пытается вычислить секрет. Если это удалось, то это значит, что $i \in \{i_1, \ldots, i_k\}$, и вычисленный секрет есть ключ K.

6.7 Генерация ЗК группой участников

Участники $B_1, \ldots, B_n$ получают от A доли секрета, по которым они могут совместно сгенерировать K^-. Участники используют несекретные числа p и a, где p – простое, $a \in \mathbf{Z}^*$ – примитивный элемент. Ниже все вычисления - в $\mathbf{Z}_p$.

1. - $A \to \text{KDC} : a^\alpha$, где $\alpha \underset{r}{\in} \mathbf{Z}_p^*$
   - участники из $\{B_1, \ldots, B_n\}$ разделяют α (используя СРС)
   - $\text{KDC} \to A : \beta \underset{r}{\in} \mathbf{Z}_p^*$ (по КП bit commitment)
   - $K^- := (\alpha + \beta) \,\%\, (p-1)$

2. Если участники $B_1, \ldots, B_n$ верят друг другу, то можно использовать следующий КП.

   - для каждого $i = 1, \ldots, n$

     $$A \to B_i : s_i \underset{r}{\in} \mathbf{Z}_p^*$$

   - $A \to \text{KDC} : t := a^s$, где

     $$s := K := (s_1 + \ldots + s_n) \,\%\, (p-1)$$

   - для каждого $i = 1, \ldots, n$

     $$B_i \to \text{KDC} : S_{B_i}(t_i), \quad \text{где } t_i := a^{s_i}$$

   - KDC признаёт K iff

     $$t = t_1 \cdot \ldots \cdot t_n$$

Можно обобщить эту задачу до задачи распределения среди участников $B_1, \ldots, B_n$ долей секрета, так, чтобы любое подмножество из $\geq m$ этих участников могло восстановить K.

6.8 Анонимная передача СК

В данном КП предполагается, что все участники используют асимметричную ШС, причём операции шифрования коммутируют, т.е. для произвольной пары участников A, B имеет место (6.1). Одним из участников является центр распределения ключей, который мы будем обозначать символом T. Его работа заключается в том, что он генерирует последовательность ключей

$$\{K_i \mid i \geq 1\}$$

и посылает зашифрованные ключи $K_T^+(K_i)$ всем участникам.

Участник A генерирует СК следующим образом.

1. A выбирает произвольный зашифрованный ключ $K_T^+(K_i)$ из той последовательности, которую посылает ему T
2. A ждёт некоторое время (чтобы затруднить установление выбранного ключа)
3. $A \to T : K_A^+(K_T^+(K_i))$
4. $T \to A : K_A^+(K_i)$
5. A использует в качестве СК ключ K_i

В результате работы этого КП никто (даже T) не будет знать, какой сеансовый ключ имеет участник A.

6.9 КП EKE генерации СК с взаимной аутентификацией

EKE = Encrypted Key Exchange.

6.9.1 1 вариант

A и B имеют общий пароль P.

- $A \to B : A, P(K_A^+)$ (шифрование на ключе P)
- $B \to A : P(K_A^+(K))$
- $A \to B : K(N_A)$
- $B \to A : K(N_A, N_B)$
- $A \to B : K(N_B)$

6.9.2 2 вариант

A и B имеют p, a, обладающие свойствами, описанными выше. Ниже все вычисления – в $\mathbf{Z}_p$.

- $A \to B : A, a^x$
- B вычисляет $K := a^{xy}$
- $B \to A : P(a^y), K(N_B)$
- $A \to B : K(N_A, N_B)$

- $B \to A : K(N_A)$

Данный КП можно усилить, заменив в нём последние 2 действия на следующие действия:

- $A \to B : K(x, N_A)$
- $B \to A : K(x, y, N_B)$
- $K' := N_A \oplus N_B$

K' является искомым сеансовым ключом.

6.9.3 3 вариант

Все вычисления делаются в $\mathbf{Z}_p$.

- $A \to B : K_B^+(a^x)$
- $B \to A : K_A^+(a^y)$
- $A \to B : K(S_A(K))$, где $K := a^{xy}$

6.10 Предварительное распределение ключевой информации

Для уменьшения объёма хранимой и передаваемой информации при генерации криптографических ключей используются **схемы распределения ключевой информации (КИ)**.

Задача распределения КИ заключается в следующем. Имеются

- некоторая симметричная шифрсистема, и
- n участников

$$A_1, \ldots, A_n \quad (n > 2) \tag{6.2}$$

Каждый участник A_i должен получить КИ $\Im_i$, используя которую, он может сгенерировать ключ K_{ij} этой ШС для связи с каждым другим участником A_j из совокупности (6.2).

Мы рассмотрим две схемы распределения КИ:

- схему Блома, и
- схему KDP (Key Distribution Patterns).

6.10.1 Схема Блома

Обозначим символом P поле из 2^l элементов, где l – длина ключей, которые должны сгенерировать участники. Мы будем рассматривать эти ключи как элементы поля P.

Сопоставим каждому участнику A_i из совокупности (6.2) некоторый элемент $r_i \in P \setminus \{0\}$, так, чтобы все элементы $r_1, \ldots, r_n$ были различны. Элементы $r_1, \ldots, r_n$ несекретны.

Для создания КИ используется симметричная матрица над P

$$A = \begin{pmatrix} a_{00} & \ldots & a_{0m} \\ \ldots & \ldots & \ldots \\ a_{m0} & \ldots & a_{mm} \end{pmatrix} \tag{6.3}$$

где $1 \leq m < n$.

КИ $\Im_i$ представляет собой строку

$$(1\ r_i\ \ldots\ r_i^m) \cdot A \tag{6.4}$$

Для любых $i, j \in \{1, \ldots, n\}$ ключ K_{ij} вычисляется следующим образом:

$$K_{ij} := (1\ r_i\ \ldots\ r_i^m) \cdot A \cdot \begin{pmatrix} 1 \\ r_j \\ \ldots \\ r_j^m \end{pmatrix} \tag{6.5}$$

Теорема 1. Раскрытие всех ключей любых m участников из совокупности (6.2) не даёт возможности вычислить ключи K_{ij}, где i и j – номера участников, ключи которых не раскрыты.

Доказательство. Мы можем считать, что те участники, все ключи которых раскрыты, имеют номера от 1 до m, т.е. известны все значения K_{ij} $(= K_{ji})$, где

$$i \in \{1, \ldots, m\}, \quad j \in \{1, \ldots, n\} \tag{6.6}$$

Пусть $i, j \in \{m+1, \ldots, n\}$. Докажем, что для любого $a \in P$ существует матрица A вида (6.3) обладающая следующими свойствами:

- значения ключей участников с номерами от 1 до m, вычисляемые по этой матрице в соответствии с правилом (6.5), совпадают с заданными известными значениями, и
- значение ключа K_{ij}, вычисляемое по этой матрице в соответствии с правилом (6.5), равно a.

Из определения (6.5) следует, что если значения ключей K_{ij} определяются в соответствии с правилом (6.5), то имеет место матричное равенство

$$K = P \cdot A \cdot Q \tag{6.7}$$

где символы K, P, Q обозначают матрицы

$$K = \begin{pmatrix} K_{11} & \dots & K_{1m} & K_{1j} \\ \dots & \dots & \dots & \dots \\ K_{m1} & \dots & K_{mm} & K_{mj} \\ K_{i1} & \dots & K_{im} & K_{ij} \end{pmatrix}$$

$$P = \begin{pmatrix} 1 & r_1 & \dots & r_1^m \\ \dots & \dots & \dots & \dots \\ 1 & r_m & \dots & r_m^m \\ 1 & r_i & \dots & r_i^m \end{pmatrix}$$

$$Q = \begin{pmatrix} 1 & \dots & 1 & 1 \\ r_1 & \dots & r_m & r_j \\ \dots & \dots & \dots & \dots \\ r_1^m & \dots & r_m^m & r_j^m \end{pmatrix}$$

Поскольку матрицы P и Q обратимы, то (6.7) эквивалентно равенству

$$A = P^{-1} \cdot K \cdot Q^{-1} \tag{6.8}$$

Следовательно, определив матрицу A по формуле (6.8), где все элементы матрицы K имеют заданные значения (в частности, $K_{ij} = a$), мы получим, что определённая таким образом матрица A обладает вышеупомянутыми свойствами. ■

Теорема 2.

Пусть схема распределения КИ между n участниками обладает свойством, изложенным в формулировке теоремы 1, и ключи, генерируемые этой схемой, состоят из l битов.

Тогда размер (в битах) КИ, которую должен иметь каждый участник, не меньше, чем $l(m + 1)$.

Отметим, что размер КИ в схеме Блома равен $l(m + 1)$, т.е. эта схема обеспечивает наиболее экономное распределение КИ, обладающее свойством, изложенным в формулировке теоремы 1.

6.10.2 Схема KDP

В данной схеме тоже имеется $n > 2$ участников. Кроме того, имеется q закрытых ключей $K_1, \dots, K_q$. Схему KDP с такими параметрами обозначают знакосочетанием

$$\mathrm{KDP}(n, q)$$

КИ каждого участника A_i представляет собой некоторое подмножество

$$\Im_i \subseteq \{K_1, \ldots, K_q\}$$

причём

- номера ключей, входящих в $\Im_i$, известны
- выполнено следующее условие: $\forall i, j, r \in \{1, \ldots, n\}$

  $$\text{если } \Im_i \cap \Im_j \subseteq \Im_r, \quad \text{то} \quad i = r \text{ или } j = r$$

  (которое эквивалентно тому, что в совокупности множеств

  $$\{\Im_i \cap \Im_j \mid 1 \leq i < j \leq n\}$$

  отсутствуют сравнимые пары, такие семейства подмножеств называются **семействами Шпернера**)

Для любых $i, j \in \{1, \ldots, n\}$ ключ K_{ij} вычисляется по формуле

$$K_{ij} := K_{i_1} \oplus \ldots \oplus K_{i_k}$$

где $\{K_{i_1}, \ldots, K_{i_k}\} = \Im_j \cap \Im_j$.

Теорема 3. Для каждого $i \in \{1, \ldots, n\}$ имеет место неравенство

$$|\Im_i| \geq \log_2 n \tag{6.9}$$

Кроме того, если $n \geq 4$, то

$$q \geq 2 \cdot \log_2 n \tag{6.10}$$

Доказательство. Если для некоторого i неравенство (6.9) не выполняется, т.е. $|\Im_i| < \log_2 n$, то имеет место неравенство $2^{|\Im_i|} < n$, т.е. $2^{|\Im_i|} \leq n - 1$, откуда следует, что количество непустых подмножеств в $\Im_i$ не превосходит $n - 2$. Этого мало: участник A_i должен иметь $n - 1$ различных ключей для связи с остальными участниками.

Неравенство (6.10) следует из

$$C_q^{[q/2]} \geq C_n^2 \tag{6.11}$$

которое, в свою очередь, следует из

$$C_q^{[q/2]} \geq n$$

причём равенство достигается iff $\{\Im_1, \ldots, \Im_n\}$ состоит из всех w–элементных подмножеств множества $\{K_1, \ldots, K_q\}$, где

$$w = \ (q - \text{чётное}) \ ? \ \frac{q}{2} \ : \ \frac{q \pm 1}{2}$$

Доказательство.

Пополним СШ $\{\Im_1, \ldots, \Im_n\}$ до максимального СШ $\Im = \{\Im_1, \ldots, \Im_m\}$. Обозначим символом w число

$$\max_{i=1,\ldots,m} |\Im_i|$$

Будем считать, что все множества из w элементов в совокупности $\Im$ имеют номера от 1 до t. Обозначим символом $\Im'$ подсовокупность $\Im_1, \ldots, \Im_t$ совокупности $\Im$.

Докажем, что $w \leq [\frac{q}{2}]$. Пусть это неверно, т.е. $w > [\frac{q}{2}]$. Заменим в $\Im$ каждое множество $\Im_i$, где $i \in \{1, \ldots, t\}$, на все его подмножества из $w-1$ элементов. Получим СШ

$$\{R_1, \ldots, R_u, \Im_{t+1}, \ldots, \Im_m\} \tag{6.12}$$

Обозначим символом $\mathcal{R}$ совокупность множеств

$$\{R_1, \ldots, R_u\}$$

Нетрудно видеть, что

- каждое из множеств $\Im_i \in \Im'$ содержит ровно w множеств из $\mathcal{R}$
- каждое множество $R_j \in \mathcal{R}$ входит в $\leq (q - w + 1)$ множеств из $\Im'$.

Лемма. $u \geq t$

Доказательство. Рассмотрим двудольный граф, множество вершин которого делится на следующие два класса:

- вершины, соответствующие множествам из $\Im'$
- вершины, соответствующие множествам из $\mathcal{R}$

Вершина $\Im_i$ соединена ребром с вершиной R_j в том и только в том случае, когда $R_i \subseteq \Im_i$.

Вычислим число рёбер в этом графе двумя способами. Получим:

$$wt \leq (q - w + 1)u$$

Отсюда следует, что

$$t([\frac{q}{2}] + 1) \leq wt \leq (q - w + 1)u \leq (q - [\frac{q}{2}])u$$

Следовательно,

$$u \geq \frac{[\frac{q}{2}] + 1}{q - [\frac{q}{2}]} t \geq t \qquad \blacksquare$$

Из леммы следует, что количество элементов в (6.12) не меньше чем m.

Если в $\Im$ есть множество с $< [\frac{q}{2}]$ числом элементов, то мы рассмотрим совокупность множеств

$$\{K \setminus \Im_1, \ldots, K \setminus \Im_m\}$$

Из данной совокупности можно построить СШ, в котором все множества имеют одинаковое число элементов, равное $[\frac{q}{2}]$.

6.11 Генерация закрытого ключа с использованием ОКС

В этом пункте рассматриваются КП, в которых участники, обмениваясь информацией по ОКС, генерируют общий закрытый ключ.

6.11.1 КП с использованием нескольких ОКС

Пусть участники A и B могут использовать n ОКС. Некоторые из этих каналов контролируются активным противником. Предполагается, что число t каналов, которые контролируются противником, не превосходит $(n-1)/2$, но номера контролируемых каналов неизвестны.

A и B хотят обмениваться зашифрованными сообщениями, для чего им необходимо сгенерировать общий ключ. Один из возможных способов генерации общего ключа имеет следующий вид.

1. Участники выбирают простое число $p > n$, так, чтобы ключ K, который должен быть сгенерирован, можно было рассматривать как элемент $\mathbf{Z}_p$.

2. A генерирует

   - ключ $K \in \mathbf{Z}_p$, и
   - полином $f \in \mathbf{Z}_p[x]$ степени t, такой, что $K = f(0)$.

3. Для каждого $i = 1, \ldots, n$
   $$A \xrightarrow{i} B : f(i)$$
   где $\xrightarrow{i}$ обозначает передачу сообщения по каналу номер i.

4. Обозначим символами $a_1, \ldots, a_n$ те сообщения, которые получил B по каналам $1, \ldots, n$ соответственно. Те сообщения, которые получены по каналам, доступным противнику, могут отличаться от соответствующих сообщений, которые посылал A.

5. B ищет полином $g \in \mathbf{Z}_p[x]$ степени t, такой, что

$$\forall i = 1, \ldots, n \quad g(i) = a_i \tag{6.13}$$

Заметим, что из неравенства $t \leq (n-1)/2$ следует неравенство $t+1 \leq n-t$. Поскольку

- количество правильно переданных сообщений $\geq n - t \geq t + 1$, и
- полином степени t однозначно определяется своими значениями в $t+1$ точке

то в том случае, когда B может построить полином g со свойством (6.13), имеет место равенство $g = f$. В этом случае B может вычислить $K := g(0)$.

Если же B не может построить полином g со свойством (6.13), то выполняются следующие действия.

- $B \to A : \{(i, a_i) \mid i = 1, \ldots, n\}$
  Это сообщение посылается по всем каналам. Возможно, что сообщения, которые получит A по разным каналам, будут различными, но количество сообщений, переданных без искажений, будет не меньше, чем $(n+1)/2$. Поэтому A может установить, какие именно сообщения переданы без искажений.
- $A \to B : \{i \mid f(i) \neq a_i\}$
  B интерпретирует присланные числа как номера некоторых каналов, которые контролируются противником.
- Весь КП повторяется сначала, с использованием лишь оставшихся КС.

После $\leq t$ итераций передача K будет успешной.

6.11.2 КП Диффи-Хеллмана

Параметры:

- p - простое, причём $\frac{p-1}{2}$ тоже простое, и
- $a \in \mathbf{Z}_p^*$ – примитивный элемент.

Числа p и a несекретны. Ниже все вычисления - в $\mathbf{Z}_p$.

- $A \to B : X := a^x$, где $x \underset{r}{\in} \mathbf{Z}_p^*$
- $B \to A : Y := a^y$, где $y \underset{r}{\in} \mathbf{Z}_p^*$

- A вычисляет $K := Y^x$
- B вычисляет $K := X^y$

Этот КП уязвим к атаке активным противником M, который заменяет пересылаемые сообщения на свои:

- M подменяет a^x и a^y на a^{x_1} и a^{y_1} соответственно, где $x_1, y_1 \underset{r}{\in} \mathbf{Z}_p^*$
- $K_{AM} := (a^x)^{y_1}, \quad K_{BM} := (a^y)^{x_1}$

6.11.3 КП Диффи-Хеллмана с ≥ 3 участниками

Мы рассмотрим КП только для 3 участников (КП для > 3 участников аналогичен).

Все параметры в данном КП - те же, что и выше, все вычисления - в $\mathbf{Z}_p$.

- $A \to B : X := a^x$, где $x \underset{r}{\in} \mathbf{Z}_p^*$
- $B \to C : Y := a^y$, где $y \underset{r}{\in} \mathbf{Z}_p^*$
- $C \to A : Z := a^z$, где $z \underset{r}{\in} \mathbf{Z}_p^*$
- $A \to B : Z' := Z^x$
- $B \to C : X' := X^y$
- $C \to A : Y' := Y^z$
- A вычисляет $K := (Y')^x$
- B вычисляет $K := (Z')^y$
- C вычисляет $K := (X')^z$

6.11.4 КП STS (station-to-station)

КП STS обеспечивает взаимное подтверждение правильности вычисления сеансового ключа K.

Все параметры в данном КП - те же, что и выше, все вычисления - в $\mathbf{Z}_p$.

- $A \to B : X := a^x$, где $x \underset{r}{\in} \mathbf{Z}_p^*$
- $B \to A : \begin{pmatrix} a^y \\ K(S_B(a^x, a^y)) \end{pmatrix}$, где $K := X^y, \ y \underset{r}{\in} \mathbf{Z}_p^*$
- $A \to B : K(S_A(a^x, a^y))$, где $K := (a^y)^x$

6.11.5 КП MTI

MTI = Мацумото-Такашима-Имаи.

Параметры:

- p – простое число, $a \in \mathbf{Z}_p^*$ – примитивный элемент
- $K_A^- = i, K_B^- = j$, где $i, j \in \mathbf{Z}_p^*$
- $K_A^+ = a^i,\ K_B^+ = a^j$

Ниже все вычисления - в $\mathbf{Z}_p$.

- $A \to B : X := a^x$, где $x \underset{r}{\in} \mathbf{Z}_p^*$
- $B \to A : Y := a^y$, где $y \underset{r}{\in} \mathbf{Z}_p^*$
- A вычисляет $K = Y^i \cdot (K_B^+)^x$
- B вычисляет $K' = X^j \cdot (K_A^+)^y$

Подмена сообщений приведёт к тому, что K и K' будут различными.

6.11.6 КП Хьюза (Hughes)

Параметры - те же, что и в КП Диффи-Хеллмана. Все вычисления - в $\mathbf{Z}_p$.

- A вычисляет $K := a^x$, где $x \underset{r}{\in} \mathbf{Z}_p^*$
- $B \to A : Y := a^y$, где $y \underset{r}{\in} \mathbf{Z}_{p-1}^*$
- $A \to B : X := Y^x$
- B вычисляет $K' := X^Z$, где $Z := y^{-1} \pmod{(p-1)}$

6.11.7 КП Татебаяши-Мацузаки-Ньюмана

В данном КП участвует доверенный посредник T.

Параметры:

- $K_T^- = \{p, q\}$, где p и q – простые числа,
- $K_T^+ = n = pq$.

Поскольку T знает p и q, то он может быстро вычислять функцию $x^{1/3}$ в $\mathbf{Z}_n$. Ниже все вычисления – в $\mathbf{Z}_n$.

КП имеет следующий вид:

- $A \to T : x^3$, где $x \underset{r}{\in} \mathbf{Z}_n$
- $B \to T : y^3$, где $y \underset{r}{\in} \mathbf{Z}_n$
- $T \to A : x \oplus y$ (побитовое сложение по mod 2)
- A вычисляет $K := (x \oplus y) \oplus x \quad (= y)$

Атака: C перехватывает сообщение y^3 на шаге 2, а затем исполняет этот КП совместно с сообщником D:

- $C \to T : y^3 u^3$, где $u \in \mathbf{Z}_n^*$
- $D \to T : v^3$
- $T \to C : (yu) \oplus v$
- $D \to C : v$
- C вычисляет $K := (((yu) \oplus v) \oplus v) \cdot u^{-1}$.

6.11.8 Генерация ключа несколькими участниками

Участники: $A_0, \ldots, A_{n-1}$. Параметры - те же, что и в КП Диффи-Хеллмана. Все вычисления - в $\mathbf{Z}_p$. Нижние индексы рассматриваются по модулю n.

Каждый участник A_i выполняет следующую последовательность действий.

- $A_i \to \{A_{i-1}, A_{i+1}\} : \quad z_i := a^{x_i}$, где $x_i \underset{r}{\in} \mathbf{Z}_p^*$
- $A_i \to$ всем : $\quad u_i := \left(\frac{z_{i+1}}{z_{i-1}}\right)^{x_i}$
- A_i вычисляет $K := z_{i-1}^{nx_i} \cdot u_i^{n-1} \cdot u_{i+1}^{n-2} \cdot \ldots \cdot u_{i+n-2}^{1}$

6.12 Квантовая передача ключей

В КП квантовой передачи ключей используются два канала связи: обычный и квантовый.

- **Обычный канал связи (ОКС)** является открытым, и противник может прослушивать все сообщения, пересылаемые по этому каналу (но не может менять их порядок, а также делать вставки или удаления). Факт чтения противником того или иного сообщения невозможно обнаружить.
- По **квантовому каналу связи (ККС)** пересылаются фотоны. При прослушивании фотонов в их параметры с большой вероятностью вносятся искажения.

6.12.1 Обработка информации в ККС

Каждому фотону, передаваемому по ККС, можно сопоставить параметр, называемый **углом поляризации**. Мы будем предполагать, что углы поляризации фотонов, передаваемых по ККС, могут иметь одно из следующих четырёх значений: 0, 45, 90 и 135 градусов. Мы будем говорить, что фотоны с такими углами поляризации имеют тип $-$, $/$, $|$ и $\backslash$ соответственно.

Для определения типа фотона используется специальный прибор, называемый **фильтром**. Из законов квантовой механики следует, что если фотон с углом поляризации α, попадает на фильтр, расположенный под углом β, то этот фотон

- пройдёт через фильтр с вероятностью $\cos^2(\alpha - \beta)$
- поглотится с вероятностью $\sin^2(\alpha - \beta)$

Мы будем говорить, что фильтр

- имеет тип $+$, если он расположен под углом 0 градусов, и
- имеет тип $\times$, если он расположен под углом 45 градусов.

Фильтр называется **совместимым** с фотоном, если

- фильтр имеет тип $+$, и фотон имеет тип $-$ или $|$, или
- фильтр имеет тип $\times$, и фотон имеет тип $/$ или $\backslash$.

Определение типа фотона при помощи фильтра производится следующим образом.

- Если фильтр имеет тип $+$, то
  - если измеряемый фотон прошёл через этот фильтр, то мы будем считать, что тип этого фотона равен $-$, и
  - если измеряемый фотон поглотился этим фильтром, то мы будем считать, что тип этого фотона равен $|$.
- Если фильтр имеет тип $\times$, то
  - если измеряемый фотон прошёл через этот фильтр, то мы будем считать, что тип этого фотона равен $/$, и
  - если измеряемый фотон поглотился этим фильтром, то мы будем считать, что тип этого фотона равен $\backslash$.

Из всего сказанного выше следует, что тип фотона определяется фильтром правильно в том и только в том случае, когда этот фильтр совместим с измеряемым фотоном.

6.12.2 КП квантовой передачи ключа

КП квантовой передачи ключа от A к B состоит из двух частей:

1. передача по ККС от A к B случайным образом сгенерированной цепочки фотонов заранее оговоренной длины n, и
2. обсуждение по ОКС.

Участник A выполняет следующие действия.

- A генерирует случайным образом слово ω длины n в алфавите $\{-, |, /, \backslash\}$

  Например, ω может иметь такой вид:

  $$| \; | \; / \; - \; - \; \backslash \; - \; | \; - \; /$$

- A создаёт цепочку из n фотонов, причём тип каждого фотона совпадает с соответствующей компонентой слова ω.
- A посылает эту цепочку фотонов участнику B.

Участник B для обработки полученной цепочки фотонов выполняет следующие действия.

- B генерирует случайным образом слово ψ длины n в алфавите $\{+, \times\}$.

  Например, ψ может иметь такой вид:

  $$\psi = \quad \times \; + \; + \; \times \; \times \; \times \; + \; \times \; + \; +$$

- B создаёт цепочку из n фильтров типа $+$ и $\times$, причём тип каждого фильтра совпадает с соответствующей компонентой слова ψ.
- B измеряет полученную цепочку фотонов при помощи своей цепочки фильтров, причём каждый полученный фотон измеряется соответствующим ему фильтром (отметим, что количество фильтров, совместимых с соответствующими им фотонами из полученной цепочки, в среднем равно $n/2$).

  Результатом измерений является слово ω' длины n в алфавите $\{-, |, /, \backslash\}$.

Далее все пересылки осуществляются по ОКС

1. $B \to A : \psi$
2. $A \to B$: номера фильтров, совместимых с соответствующими им фотонами из ω
   (в нашем примере - это 2, 6, 7, 9)

3. A оставляет в ω только те компоненты, номера которых A послал B в предыдущем действии,
   в нашем примере это будет выглядеть так:

   $$* \mid * * * \setminus - * - *$$

   (символ $*$ обозначает удаляемые компоненты)

4. B делает то же самое с компонентами ω'

5. A заменяет

   - компоненты $-$ и $/$ в оставшейся части последовательности ω на бит 1, и
   - компоненты $\mid$ и $\setminus$ – на бит 0.

   Получившуюся последовательность битов обозначим символом K.

   В нашем примере K имеет вид

   $$(0, 0, 1, 1)$$

6. B делает те же операции со своей оставшейся частью последовательности ω'. Получившуюся последовательность битов обозначим символом K'.

Затем проверяется, было ли подслушивание в ККС:

1. $B \to A$: последовательность пар вида $(i, K'[i])$, где

   - $K'[i] = i$–й бит K',
   - количество пар в посылаемой последовательности приблизительно равно $1/3$ длины K',
   - номера тех битов K', которые включаются в посылаемую последовательность, выбираются случайным образом.

2. $A \to B : \begin{cases} 0, \text{если среди принятых пар есть пара} \\ \quad (i, K'[i]), \text{такая, что } K'[i] \neq K[i] \\ 1, \text{иначе} \end{cases}$

Если A послал B значение 1, то это означает, что подслушивания в ККС не было, и в качестве искомого ключа участники A и B могут использовать последовательности из тех компонентов K и K', которые не участвовали в последних проверках.

Если же A послал B значение 0, то это означает, что в ККС имело место подслушивание.

Этот КП можно улучшить так, чтобы даже в случае обнаружения подслушивания A и B могли бы сгенерировать общий закрытый ключ.

Глава 7

Схемы распределения секрета

7.1 Понятие СРС

Схема распределения секрета (СРС) – это способ распределения информации между несколькими участниками, используя которую, они могут вычислить некоторое заданное значение s (называемое **секретом**). Информация, которую при этом получает каждый участник, называется **долей** этого участника. Например,

- s может представлять собой вектор $s \in \{0,1\}^k$, и
- долями являются вектора $s_1, \ldots, s_n$ из $\{0,1\}^k$, такие, что выполняется соотношение

$$s = s_1 \oplus \ldots \oplus s_n$$

В данном случае значение s могут вычислить только все участники совместно, а если доля хотя бы одного из участников неизвестна, то все остальные участники, даже открыв друг другу свои доли, не могут извлечь из них никакой информации о значении s.

7.2 (n,k)-пороговая СРС

Наиболее часто используются (n,k)-пороговые СРС (где $1 < k \leq n$), в которых секрет распределяется между n участниками таким образом, что

- любая совокупность из $\geq k$ участников может, используя свои доли, вычислить секрет s, и
- любая совокупность из $< k$ участников не сможет извлечь из своих долей никакой информации о секрете s.

Примеры (n,k)-пороговых СРС:

1. СРС Шамира.

   Пусть секрет s можно представить в виде элемента некоторого поля P. Для распределения секрета выбирается многочлен $f \in P[x]$ степени $k-1$:

   $$f = a_0 + a_1 \cdot x + \ldots + a_{k-1} \cdot x^{k-1}$$

   в котором $a_0 = s$.

   Для каждого из n участников выбирается несекретный элемент $r_i \in P \setminus \{0\}$, причём элементы $r_1, \ldots, r_n$ различны.

   Доля участника $A_i = f(r_i)$.

   Для вычисления секрета используется интерполяционная формула Лагранжа: для каждого k–элементного подмножества $\{s_1, \ldots, s_k\} \subseteq \{r_1, \ldots, r_n\}$ имеет место равенство

   $$f(x) = \sum_{i=1}^{k} f(s_i) \cdot \prod_{j \neq i} \frac{x - s_j}{s_i - s_j}$$

   из которого следует, что

   $$a_0 = f(0) = \sum_{i=1}^{k} f(s_i) \cdot \prod_{j \neq i} \frac{s_j}{s_j - s_i}$$

   Если доли вычисляются и распределяются участником T, которому участники $A_1, \ldots, A_n$ не доверяют и хотят проверять правильность своих долей, то в том случае, когда $P = \mathbf{Z}_p$, это можно сделать следующим образом. Выбирается несекретный элемент $g \in P$, и

   $$T \to A_1, \ldots, A_n : \{b_i := g^{a_i} \mid i = 0, \ldots, k-1\}$$

   Участник A_i, после получения своей доли $\delta_i = f(r_i)$, проверяет её правильность следующим образом:

   $$g^{\delta_i} = b_0 \cdot b_1^{r_i} \cdot \ldots \cdot b_{k-1}^{(r_i^{k-1})}$$

2. Векторная СРС (Blakley).

   $s \in \mathbf{R}^k$, каждая доля представляет собой уравнение $(k-1)$–мерной гиперплоскости, содержащей s. $\{s\}$ = пересечение любых k долей.

3. СРС Асмута-Блума. В этой СРС $s \in \mathbf{Z}$.

   Выбираются простое число $p > s$, и n взаимно простых чисел $d_1 < \ldots < d_n$, таких, что

   $$d_1 \cdot \ldots \cdot d_k > p \cdot \underbrace{d_{n-k+2} \cdot \ldots \cdot d_n}_{k-1}$$

   Доля участника i есть число $(s + ap) \ \% \ d_i$, где $a \underset{r}{\in} \mathbf{Z}$. Для вычисления s используется китайская теорема об остатках.

4. СРС Карнина-Грини-Хеллмана. Здесь $s \in \mathbf{R}$.

   Выбираются вектора $\vec{v}_0, \ldots, \vec{v}_n \in \mathbf{R}^k$, такие, что ранг матрицы из любых k этих векторов равен k. Например,

   $$\vec{v}_i = (1, r_i, r_i^2, \ldots, r_i^{k-1})$$

   Секрет s имеет вид

   $$s := \vec{v}_0 \cdot u^{\downarrow}, \quad \text{где } u^{\downarrow} \underset{r}{\in} \mathbf{R}^k$$

   Доля i–го участника ($i = 1, \ldots, n$) имеет вид $\vec{v}_i \cdot u^{\downarrow}$. Каждые k долей порождают систему линейных уравнений размера $k \times k$, неизвестными в которой являются коэффициенты вектора $u^{\downarrow}$.

7.3 СРС с недоверяющими участниками

Предполагается, что совокупность участников состоит из двух групп, и участники из разных групп не доверяют друг другу.

Для вычисления секрета s нужно

- m_1 любых долей участников из первой группы, и
- m_2 любых долей участников из второй группы.

Создаётся пара многочленов f, g, где

- степень f равна $m_1 - 1$, и
- степень g равна $m_2 - 1$.

Каждый участник из первой группы получает $f(x_i)$, и каждый участник из второй группы получает $g(y_j)$.

Многочлен, по которому вычисляется секрет, равен произведению $f \cdot g$.

7.4 Контроль правильности долей в СРС Шамира

$P = \mathbf{Z}_p$

$s := f_1(0) + f_2(0)$, где

- $f_1(x) = a_0 + \ldots + a_{k-1}x^{k-1}$
- $f_2(x) = b_0 + \ldots + b_{k-1}x^{k-1}$

$\forall i = 0, \ldots, k-1$

$$r_i^{(1)} = g^{a_i}, \quad r_i^{(2)} = g^{b_i}$$

$\forall i = 1, \ldots, n$ A_i получает $f_1(i)$ и $f_2(i)$, его доля $= f_1(i) + f_2(i)$.

Проверка: $g^{f_1(i)+f_2(i)} = r_i^{(1)} \cdot r_i^{(2)}$

7.5 Структура доступа

Структура доступа (СД) на множестве $\{1, \ldots, n\}$ – это совокупность Γ некоторых подмножеств множества $\{1, \ldots, n\}$ такая, что

- вычислить s могут только подмножества из Γ
- подмножества не из Γ не могут получить никакой информации о секрете

7.6 Линейная СРС

Пусть L – линейное пространство над полем P.

Линейная реализация СД Γ – это совокупность $\mathcal{L}$ из $n+1$ подпространств пространства L вида

$$\mathcal{L} = \{L_0, \ldots, L_n\}$$

удовлетворяющая условию:

- $\forall I \in \Gamma \quad L_0 \subseteq L_I$, где $L_I \stackrel{\text{def}}{=} \langle \bigcup\limits_{i \in I} L_i \rangle$
- $\forall I \notin \Gamma \quad L_0 \cap L_I = \{0\}$

Глава 8

Доказательства с нулевым разглашением

8.1 Понятие Д0Р

Один из классов КП состоит из протоколов, называемых **доказательствами с нулевым разглашением (Д0Р)**. Данные КП используются в тех случаях, когда один из участников (обозначаемый обычно символом P, от слова prover) хочет убедить другого участника (обозначаемого обычно символом V, от слова verifier), что он знает некоторый секрет, которым может быть

- некоторое число, или
- знание доказательства некоторого утверждения (например, утверждения вида $a \in L$, где L – некоторый NP-полный язык)

таким образом, чтобы V

- убедился в том, что P знает секрет,
- но при этом не смог извлечь из сообщений, которые он получает от P в процессе функционирования КП, никакой информации об этом секрете.

Д0Р может быть интерактивным и неинтерактивным.

Интерактивное Д0Р представляет собой совокупость из нескольких раундов обмена сообщениями между P и V, где каждый раунд имеет вид

- $V \to P$: вопрос$_i$
- $P \to V$: ответ$_i$
- V проверяет правильность полученного ответа.

Если P даёт правильный ответ на каждый вопрос, который задаёт ему V в процессе функционирования КП, то V принимает решение, что P знает секрет.

Вопросы, которые V задаёт P, как правило, представляют собой выбранные случайным образом элементы некоторого множества. Задача поиска ответов на вопросы должна обладать следующими свойствами.

1. Если P действительно знает секрет, то он может за полиномиальное время найти правильный ответ на каждый вопрос, который он получит от V.

2. Если P не знает секрета, то вероятность того, что ему удастся найти правильный ответ на произвольный вопрос от V за полиномиальное время, должна быть ограничена некоторым числом q, где $0 \leq q < 1$.

   Из этого условия следует, что P сможет найти правильные ответы во всех n раундах, в каждом из которых V генерирует свой вопрос независимо от других раундов, с вероятностью, не превышающей q^n. За счёт выбора n можно добиться того, чтобы P, не зная секрета, мог убедить V в том, что он знает секрет (т.е. смог найти правильные ответы на все заданные ему вопросы), можно сделать сколь угодно малой.

Неинтерактивное Д0Р представляет собой однократную посылку сообщения

$$P \to V : \text{ доказательство}$$

Как правило, неинтерактивное Д0Р представляет собой запись раундов обычного Д0Р, в которых вопросы V к P генерируются самим P с использованием некоторой хэш-функции (что даёт основания рассматривать данные вопросы как сгенерированные случайным образом).

Класс Д0Р включает в себя многие КП аутентификации и ПЭП.

8.2 Д0Р знания изоморфизма графов

Пусть k – положительное целое число. Обозначим символом N_k множество $\{1, \ldots, k\}$. Мы будем считать, что множество вершин каждого графа, рассматриваемого в настоящем пункте, имеет вид N_k для некоторого $k \geq 1$.

Для

- каждого графа G с множеством вершин N_k, и
- каждой перестановки ξ на N_k (т.е. биективного отображения $\xi : N_k \to N_k$)

знакосочетание $\xi(G)$ обозначает граф с множеством вершин N_k, множество рёбер которого определяется следующим образом: для любых $i, j \in N_k$

$$G \text{ содержит ребро } i \to j \text{ тогда и только тогда, когда}$$
$$\xi(G) \text{ содержит ребро } \xi(i) \to \xi(j)$$

Графы G_1 и G_2 называются **изоморфными**, если $G_1 = \xi(G_2)$ для некоторой перестановки ξ. Знакосочетание $G_1 \sim G_2$ обозначает утверждение о том, что G_1 и G_2 изоморфны.

Если P знает доказательство того, что $G_1 \sim G_2$ (т.е. P знает перестановку σ, такую, что $G_1 = \sigma(G_2)$), то P может доказать V то, что он знает такую σ при помощи Д0Р. Один раунд этого Д0Р имеет следующий вид.

1. $P \to V : \xi(G_1)$, где ξ – случайным образом порождённая перестановка на множестве вершин G_1
2. $V \to P : i \underset{r}{\in} \{1, 2\}$
3. $P \to V$: перестановка, доказывающая $\xi(G_1) \sim G_i$
   т.е. $P \to V : (i = 1)\ ?\ \xi : \xi \circ \sigma$

8.3 Д0Р знания существования гамильтонова цикла в графе

Гамильтонов цикл (ГЦ) в графе - это цикл в этом графе (т.е. путь, начало которого совпадает с его концом), который проходит через каждую вершину ровно один раз.

Если P знает некоторый ГЦ в графе G, то P может убедить в этом V при помощи Д0Р. Один раунд этого Д0Р имеет следующий вид.

1. P случайным образом порождает перестановку ξ на множестве вершин графа G
2. $P \to V : H$, где H – матрица, каждый элемент которой представляет собой результат вероятностного шифрования соответствующего элемента матрицы инцидентности графа $\xi(G)$ (который равен 0 или 1)
3. $V \to P : i \underset{r}{\in} \{1, 2\}$
4. $P \to V : (i = 1)\ ?\ \xi : \sigma$, где σ – список элементов матрицы H, результат расшифровки которых представляет собой ГЦ в графе $\xi(G)$

8.4 Общая схема Д0Р

Приведённые выше Д0Р можно рассматривать как частные случаи общей схемы, позволяющей доказывать с нулевым разглашением знание решения некоторого класса задач.

Предполагается, что на множестве задач $\mathcal{A}$ из этого класса задана некоторая совокупность Ξ преобразований, обладающих следующими свойствами.

- Каждое преобразование $\xi : \mathcal{A} \to \mathcal{A}$ сопоставляет каждой задаче $a \in \mathcal{A}$ задачу $\xi(a) \in \mathcal{A}$, изоморфную (в некотором смысле) задаче a.
- Если участник знает решение σ некоторой задачи $a \in \mathcal{A}$, то для любого $\xi \in \Xi$ данный участник знает решение задачи $\xi(a)$. Мы будем обозначать это решение знакосочетанием $\xi(\sigma)$.

Если P знает решение σ задачи a, то P может доказать V то, что он знает σ, при помощи Д0Р. Один раунд этого Д0Р имеет следующий вид.

1. $P \to V : (\xi(a), [\xi(\sigma)])$, где ξ – выбранное случайным образом преобразование из Ξ. $([\cdot] = \text{bit commitment})$
2. $V \to P : i \underset{r}{\in} \{1, 2\}$
3. $P \to V : (i = 1)\ ?\ \xi : \xi(\sigma)$

Данное интерактивное Д0Р можно преобразовать в неинтерактивное Д0Р следующим образом.

1. P выбирает случайным образом преобразования $\xi_1, \ldots, \xi_n \in \Xi$.
2. $P \to V : \begin{cases} \{(\xi_i(a), [\xi_i(\sigma)]) \mid i = 1, \ldots, n\} \\ \{(b_i = 0)\ ?\ \xi_i : \xi_i(\sigma) \mid i = 1, \ldots, n\} \end{cases}$

где $b_1, \ldots, b_n$ – первые n битов (которые можно рассматривать как случайные) строки $h(m)$, где

- h - хэш-функция, и
- $m = \{(\xi_i(a), [\xi_i(\sigma)]) \mid i = 1, \ldots, n\}$

8.5 Д0Р знания ЗК в RSA

Пусть P использует систему RSA с модулем n, ОК e и ЗК d. Если P хочет убедить V, в том, что он знает d, без раскрытия значения d, то он может сделать это при помощи Д0Р. Ниже все вычисления и сравнения производятся по $\bmod n$.

Сначала P и V

- совместно генерируют число $k \in \{4, \ldots, n-1\}$, используя КП подбрасывания монеты, и
- вычисляют $m \in \{4, \ldots, n-1\}$, такое, что $k \cdot m = e$
  (если такого m не существует, то k генерируется ещё раз).

Один раунд Д0Р имеет следующий вид.

- P и V совместно генерируют $u \in \mathbf{Z}_n$, используя КП подбрасывания монеты
- $P \to V : x := (u^d)^k$
- V принимает ответ iff $x^m = u$.

8.6 Д0Р знания дискретного логарифма

Пусть a, b, p – несекретные числа, такие, что

- p – простое, и
- существует число $x \in \mathbf{Z}^*_{p-1}$, такое, что $a^x \underset{p}{=} b$.

Если P знает такое число x, то он может убедить в этом V при помощи следующего КП. Ниже все вычисления и сравнения производятся по $\bmod p$.

1. $P \to V : \{h_i := a^{r_i} \mid i = 1, \ldots, t\}$, где для каждого $i = 1, \ldots, t$
$$r_i \underset{r}{\in} \{1, \ldots, p-2\}$$
2. P и V совместно генерируют биты $b_1, \ldots, b_t$, используя КП подбрасывания монеты
3. $j :=$ min число из $\{1, \ldots, t\}$ со свойством $b_j = 1$
4. $P \to V : \{(b_i = 0)\ ?\ r_i : s_i \mid i = 1, \ldots, t\}$, где для каждого $i = 1, \ldots, t$
$$s_i := (r_i - r_j)\%(p-1)$$

5. $\forall\, i = 1, \ldots, t$ участник V проверяет, что

$$(b_i = 0)\ ?\ (a^{r_i} = h_i) : (a^{s_i} = h_i \cdot h_j^{-1})$$

6. $P \to V : z := (x - r_j)\%(p - 1)$

7. V принимает ответ iff $a^z = b \cdot h_j^{-1}$

Можно доказать, что

- данный КП является Д0Р, и
- вероятность удачного мошенничества со стороны участника P не превосходит 2^{-t}.

Глава 9

Совместная генерация случайных значений

Данный вид КП обычно называют **КП подбрасывания монеты**. Целью таких КП является совместная генерация участниками последовательности случайных битов. В настоящей главе мы рассмотрим несколько способов решения этой задачи.

9.1 Закрытая передача

Для того, чтобы сгенерировать последовательность битов заданной длины n, участники A и B могут каждый по отдельности сгенерировать последовательности u и v длины n, и определить искомую последовательность как покомпонентную сумму u и v по модулю 2. Для вычисления этой суммы A и B должны послать друг другу свои последовательности. Тот из участников, которым первым посылает свою последовательность, должен быть уверен, что после этой передачи его партнёр не изменит свою последовательность так, чтобы получить желательный для него результат.

Предположим для определённости, что первым посылает свою последовательность участник A. Он должен послать не саму последовательность u, а такую строку $[u]$ (которую мы будем называть **закрытой передачей** строки u), чтобы

- по $[u]$ было бы невозможно вычислить u, и
- по u и $[u]$ можно было бы установить (возможно, с использованием дополнительной информации), действительно ли $[u]$ была получена из u.

Обмен последовательностями u и v при помощи закрытой передачи можно можно реализовать, например, с использованием симметричной криптографии:

1. $B \to A : N$
2. $A \to B : [u] := K(u, N)$
3. $B \to A : v$
4. $A \to B : K$

Другой способ закрытой передачи использует хэш-функцию:

1. $A \to B : N,\ [u] := h(N, N', u)$
2. раскрытие: $A \to B : N', u$

9.2 КП bit committment

Целью данного КП является закрытая передача одного бита b (это называется **bit committment**).

Участники выбирают простые p, q, и примитивный элемент $g \in \mathbf{Z}_p^*$. Все вычисления и сравнения выполняются по $\bmod p$.

1. $B \to A : y := g^x$, где $x \underset{r}{\in} \mathbf{Z}_p$
2. $A \to B : [b] := y^b g^k$, где $k \underset{r}{\in} \mathbf{Z}_p$
3. раскрытие: $A \to B : b, k$

9.3 КП подбрасывания монеты

9.3.1 ППМ с использованием ОФ

В данном КП используется ОФ $f : \mathbf{Z} \to \mathbf{Z}$

1. $A \to B : y := f(x)$, где $x \underset{r}{\in} \mathbf{Z}$, f – ОФ
2. $B \to A$: догадка (x – чётное или нечётное)
3. $A \to B : x$

Дырки:

- если A найдёт нечётное x_1 и чётное x_2, такие, что $f(x_1) = f(x_2)$, то он сможет смошенничать
- если некоторый бит $f(x)$ коррелирует с x, то B может правильно угадать с вероятностью $> \frac{1}{2}$.

9.3.2 ППМ с использованием асимметричной ШС

В данном КП участники A и B используют асимметричную шифрсистему со свойством

$$\forall m \quad K_A^+(K_B^+(m)) = K_B^+(K_A^+(m))$$

1. $B \to A : K_B^+(N_0), K_B^+(N_1)$
   (последний бит одного из нонсов равен 0, а другого - 1)
2. $A \to B : K_A^+(K_B^+(N_i))$, где $i \underset{r}{\in} \{0,1\}$
3. $B \to A : K_B^-(K_A^+(K_B^+(N_i)))) \quad (= K_A^+(N_i))$
4. $A \to B : N_i$
5. A и B открывают свои ЗК.

9.3.3 ППМ с использованием ОФ x^2

- $A \to B : n := pq$, где p и q – большие простые
  (ниже все вычисления и сравнения - в $\mathbf{Z}_n$)
- $B \to A : [u],\ v := u^2$, где $u \underset{r}{\in} \{1, \ldots, \frac{n-1}{2}\}$
- A, зная p и q, решает уравнение в $\mathbf{Z}_n$

  $$x^2 = v$$

  которое имеет 4 решения: $a, -a, b, -b \in \mathbf{Z}_n$. Из определения u следует, что

  - $u = a' := \min\{a, -a\}$, или
  - $u = b' := \min\{b, -b\}$
- $A \to B : p, q$, догадка: $u = a'$ или b'?
- $B \to A : u$

9.3.4 ППМ с использованием ОФ x^y

Параметры:

- простое p, такое, что существует большое простое $q \mid (p-1)$
- примитивные элементы $a, b \in \mathbf{Z}_p^*$

Эти параметры можно использовать для бросания нескольких монет. Ниже все вычисления и сравнения производятся по mod p.

КП имеет следующий вид:

- $A \to B$: $y \underset{r}{\in} \{a^x,\ b^x\}$, где $x \underset{r}{\in} \mathbf{Z}^*_{p-1}$
- $B \to A$: догадка: $y = a^x$ или $y = b^x$
- $A \to B$: x

9.3.5 ППМ с использованием чисел Блюма

В этом КП n – некоторое число Блюма, т.е. имеет вид $p^r q^s$, где p и q – простые, а r и s – нечётные. Ниже все вычисления и сравнения - по mod n.

1. $A \to B$: $n, b := a^2$, где $a := x^2$, $x \underset{r}{\in} \mathbf{Z}^*_n$
2. $B \to A$: догадка: a - чётное или нет?
3. $A \to B$: x, простые множители n

Глава 10

КП голосования

10.1 Понятие КП голосования

Задача **голосования** заключается в том, что несколько участников должны совместно выбрать решение из некоторого множества возможных решений. Каждый участник заполняет свой бюллетень, отражающий решение этого участника. Совместное решение вырабатывается путём обработки всех бюллетеней.

Например, в качестве данного решения может выступать избрание некоторого лица на какую-либо должность. В этом случае каждый бюллетень представляет собой список возможных кандидатов, и заполнение бюллетеня участником заключается в том, что участник оставляет в своём бюллетене только того кандидата, которого он хотел бы видеть избранным на эту должность. Обработка бюллетеней заключается в подсчёте голосов, поданных за каждого кандидата. Избранным считается тот кандидат, за которого подано наибольшее количество голосов.

Часто при процедуре голосования важно обеспечить конфиденциальность решений, принимаемых участниками, а также некоторые другие условия, связанные с контролем правильности подсчёта голосов. Данная задача решается при помощи специальных КП, которые называются **КП голосования**.

Наиболее часто используются такие КП голосования, в которых каждый участник отсылает свой бюллетень некоторому доверенному участнику, называемому **Центральной Избирательной Комиссией (ЦИК)**, которого мы будем обозначать ниже символом T. ЦИК должен обработать полученные от участников бюллетени, и опубликовать результат голосования.

Условия на процедуру голосования, которые должен обеспечить КП голосования, могут иметь, например, следующий вид.

1. Голосовать могут только те участники, которые имеют на это право (такие участники называются **избирателями**).

2. Каждый избиратель может голосовать только один раз (т.е. может послать только один бюллетень).

3. Невозможно установить, за кого проголосовал каждый избиратель.

4. Невозможно сделать дубликат заполненного бюллетеня.

5. Невозможно изменить результат голосования каждого избирателя.

6. Каждый избиратель может проверить, что его бюллетень учтён.

7. Всем известно, кто участвовал в голосовании.

10.2 Примеры КП голосования

1. Самый простой КП голосования может заключаться в следующем: каждый избиратель A_i создаёт свой бюллетень a_i, шифрует его и посылает T:

$$A_i \to T : K_T^+(a_i)$$

T вычисляет результат и публикует его.

В данном КП не выполняются почти все вышеперечисленные условия.

2. В другом КП каждый избиратель A_i создаёт свой бюллетень a_i, подписывает его, шифрует, и посылает T:

$$A_i \to T : K_T^+(S_{A_i}(a_i))$$

после чего T вычисляет результат и публикует его.

В данном КП условия 1 и 2 выполняются, условие 3 не выполняется, условие 4 выполняется, и т.д.

10.3 КП голосования с использованием слепой ЭП

Предположим, что целью голосования является выбор одной из двух альтернатив. Каждый избиратель A_i должен иметь два бюллетеня (каждый из которых соответствует некоторой альтернативе), которые подписаны T. Избиратель A_i должен выбрать один из этих бюллетеней, и послать его T. Победившей считается та альтернатива, за которую подано больше голосов.

В излагаемом ниже КП взаимодействие избирателя A_i с T осуществляется следующим образом. В каждой из нижеследующих пересылок вида $A_i \to T$ получатель T не должен знать имени отправителя A_i.

1. A_i генерирует 10 кортежей вида

$$\{(a_j^{(1)},\ a_j^{(2)},\ N_j) \mid j = 1, \ldots, 10\}$$

где

- $a_j^{(1)}$ – j-й бюллетень участника A_i, соответствующий первой альтернативе
- $a_j^{(2)}$ – j-й бюллетень участника A_i, соответствующий второй альтернативе
- N_j нонс, который в данном случае рассматривается как один из уникальных номеров участника A_i

2. $A_i \to T : \{\left(\xi_j \cdot a_j^{(1)},\ \xi_j \cdot a_j^{(2)},\ \xi_j \cdot N_j\right) \mid j = 1, \ldots, 10\}$ где ξ_j – маскирующие множители, причём операция умножения на маскирующий множитель коммутирует с функцией вычисления ЭП S_T

3. T проверяет в своей базе данных, что раньше от A_i не было сообщений, и заносит имя A_i в свою базу данных

4. T выбирает 9 сообщений из полученных 10, просит у A_i их маскирующие множители, и открывает их

5. если все сообщения корректны, T подписывает в невскрытом кортеже $\left(\xi_{j_0} \cdot a_{j_0}^{(1)},\ \xi_{j_0} \cdot a_{j_0}^{(2)},\ \xi_{j_0} \cdot N_{j_0}\right)$ все 3 компоненты, и посылает результат A_i:

$$T \to A_i : \left(S_T(\xi_{j_0} \cdot a_{j_0}^{(1)}),\ S_T(\xi_{j_0} \cdot a_{j_0}^{(2)}),\ S_T(\xi_{j_0} \cdot N_{j_0})\right)$$

6. A_i удаляет маскирующий множитель, выбирает нужный подписанный бюллетень $S_T(a_{j_0}^{(\cdot)})$, и посылает его T вместе с подписанным номером:

$$A_i \to T : K_T^+\left(S_T(a_{j_0}^{(\cdot)}),\ S_T(N_{j_0})\right)$$

7. T проверяет

   - подлинность ЭП, и
   - уникальность номера (если номер новый, то заносит его в свою БД)

   после чего вычисляет результат, и публикует

   - результат голосования, и
   - номера голосовавших, вместе с волеизъявлением каждого номера.

Недостаток: нечестный T может фальсифицировать выборы, генерируя подписанные и достоверные бюллетени, и посылая их самому себе.

10.4 Числовой КП голосования

Параметры:

- p – простое число
  (ниже все вычисления и сравнения – в $\mathbf{Z}_p$)
- $g, h \underset{r}{\in} \mathbf{Z}_p^*$, причём $\langle g \rangle = \langle h \rangle$
- $K_T^- := x \underset{r}{\in} \mathbf{Z}_p^*, \quad K_T^+ = y := g^x$

Бюллетень избирателя A_i имеет вид $a_i := h^{b_i}$, где b_i = голос A_i, который должен быть неизвестен даже T.

Результат голосования = $z = \sum\limits_{i=1}^{n} b_i$.

КП имеет следующий вид.

1. $A_i \to T$: $\begin{cases} P_i := g^{\delta_i}, \quad \text{где } \delta_i \underset{r}{\in} \mathbf{Z}_p \\ Q_i := y^{\delta_i} \cdot a_i \end{cases}$
2. T ищет z из уравнения $h^z := \prod a_i$, где $a_i := Q_i / P_i^x$
   (если количество возможных значений для z невелико, то z ищется простым перебором)

Критерий правильности вычисления z:

$$Q = P^x$$

где $P := \prod P_i$, $Q := (\prod Q_i)/h^z$.

Для доказательства $Q = P^x$ без раскрытия значения x можно использовать интерактивный КП Шаума-Педерсена, один раунд которого имеет следующий вид.

- $T \to V : (\alpha, \beta) := (g^k,\ P^k)$, где $k \underset{r}{\in} \mathbf{Z}_p^*$
- $V \to T : e \underset{r}{\in} \mathbf{Z}_p^*$
- $T \to V : s :=\ k + e \cdot x$
- V принимает ответ iff $\begin{cases} g^s = \alpha \cdot y^e \\ P^s = \beta \cdot Q^e \end{cases}$

(T доказывает V, что $\log_P Q = \log_g y$).

10.5 Голосование с ЦИК + ЦУР

Для того, чтобы противодействовать возможной нечестности со стороны ЦИК, можно использовать второго доверенного посредника, называемого **Центральным Управлением Регистрации (ЦУР)**, и обозначаемого символом T'. Необходимым условием корректности нижеследующего КП является отсутствие обмена информацией между ЦУР и ЦИК.

Взаимодействие между A_i, T и T' осуществляется следующим образом. В каждой из нижеследующих пересылок пересылаемые сообщения должны быть подписаны и зашифрованы ОК получателя.

1. $A_i \to T'$: просьба дать регистрационный номер
2. $T' \to A_i : N_i$ (случайный рег. номер)
3. $T' \to T : \mathcal{N}$:= список всех выданных рег. номеров
4. $A_i \to T : (ID_i, N_i, a_i)$, где ID_i – случайный идентификатор избирателя A_i
5. T проверяет: $N_i \in \mathcal{N}$? Если это верно, то
   - $\mathcal{N} := \mathcal{N} \setminus \{N_i\}$
   - $\Im := \Im \sqcup \{ID_i\}$ (в начале работы $\Im = \emptyset$)
6. после получения всех бюллетеней T публикует результат, и список записей вида $(ID_i,\ a_i)$.
7. T' публикует список зарегистрированных A_i.

Если $T = T'$, то надо модифицировать этап 2: рег. номера надо выдавать анонимно, используя КП ANDOS. Если T нечестен, и может выдавать рег. номера неправомочным избирателям, то этот КП надо модифицировать с помощью слепых ЭП.

10.6 Улучшенный КП голосования

Данный КП удовлетворяет условиям 1 - 6, а также обладает следующим свойством: если избиратель A_i обнаружит, что его бюллетень был заполнен или обработан неправильно, то он может переголосовать.

Подготовка:

1. T публикует список всех участников, имеющих право голосовать
2. $A_i \to T$: намерение голосовать

3. T публикует список избирателей, собирающихся принять участие в выборах
   (это мешает T включать поддельные бюллетени)

Голосование:

1. $T \to [A_i] : ID_i$ (идентификационный номер)
2. $[A_i] \to T : \ \left(ID_i,\ K_i^+(ID_i, a_i)\right)$
3. T публикует $K_i^+(ID_i, a_i)$ (это позволяет A_i проверить, что T корректно учёл его a_i)
4. $A_i \to T : ID_i, K_i^-$
5. T расшифровывает бюллетени и обрабатывает их.
6. T публикует результаты голосования, и все
$$a_i,\ K_i^+(ID_i, a_i)$$
7. Если A_i обнаружил, что его a_i учтён неверно, то
$$A_i \to T : \ \left(ID_i,\ K_i^+(ID_i, a_i),\ K_i^-\right)$$
8. Если A_i хочет изменить выбор с a_i на a_i', то
$$A_i \to T : \ \left(ID_i,\ K_i^+(ID_i, a_i'),\ K_i^-\right)$$

Если на этапе 1 голосования T обнаруживает, что два избирателя получили одинаковые ID, то T

- генерирует новый ID',
- выбирает одного из избирателей (=: A_i) с этим ID, и
- публикует $ID', K_i^+(ID, a_i)$

A_i узнаёт о путанице, и повторно отсылает свой a_i (этап 2 голосования) с новым ид. номером ID'.

Недостатки: преступная ЦИК может

1. воспользоваться бюллетенями избирателей, которые зарегистрировались, но не голосовали
2. безнаказанно "потерять" некоторые бюллетени

10.7 Выборы без T

Для простоты мы рассмотрим случай, когда в голосовании участвуют 4 избирателя: A_1, A_2, A_3, A_4. Все избиратели используют одну и ту же асимметричную ШС. Каждый избиратель A_i имеет свой ОК K_i^+, ЗК K_i^-, и использует ЭП S_i.

1. $\forall i\ A_i \to A_1 : m_i$, где

$$\begin{aligned} m_i &:= K_1^+(\alpha_i, N_{i1}) \\ \alpha_i &:= K_2^+(\beta_i, N_{i2}) \\ \beta_i &:= K_3^+(\gamma_i, N_{i3}) \\ \gamma_i &:= K_4^+(\delta_i, N_{i4}) \\ \delta_i &:= K_1^+(\varphi_i) \\ \varphi_i &:= K_2^+(\psi_i) \\ \psi_i &:= K_3^+(\eta_i) \\ \eta_i &:= K_4^+(a_i, N_{i5}) \end{aligned}$$

2. $A_1 \to A_2$: перетасованный набор α_i
3. $A_2 \to A_3$: перетасованный набор β_i,
4. $A_3 \to A_4$: перетасованный набор γ_i
5. $A_4 \to A_1$: перетасованный набор δ_i
6. $A_1 \to A_2, A_3, A_4$: перетасованный набор $S_1(\varphi_i)$
7. $A_2 \to A_1, A_3, A_4$: перетасованный набор $S_2(\psi_i)$
8. $A_3 \to A_1, A_2, A_4$: перетасованный набор $S_3(\eta_i)$
9. $A_4 \to A_1, A_2, A_3$: перетасованный набор $S_4(a_i, N_{i5})$

Каждый раз, когда избиратель получает набор сообщений, одной из компонентов которых является нонс, он проверяет наличие среди полученных сообщений такого, в котором присутствует его нонс.

Глава 11

Электронная коммерция

11.1 КП электронной коммерции

В КП **электронной коммерции (ПЭК)** некоторые взаимодействия между участниками имеют коммерческий характер. Сообщения, передаваемые в таких взаимодействиях, являются электронными аналогами обычных бумажных денег или денежных переводов. Данные виды сообщений называются соответственно **электронными банкнотами (ЭБ)** и **электронными денежными переводами (ЭДП)**.

Свойства, которыми могут обладать ЭБ и ЭДП, могут иметь, например, следующий вид.

1. По ЭБ и ЭДП невозможно вычислить тех участников, которые использовали их в своих платежах (анонимность).
2. ЭБ можно передавать другим участникам, которые могут использовать их по своему усмотрению.
3. Любую ЭБ можно разделить в любой заданной пропорции.
4. ЭБ и ЭДП невозможно использовать более одного раза путём изготовления дубликата.
5. В момент обработки ЭБ продавцу не обязательно соединяться с банком, он может сделать это позднее.

Ниже в этой главе некоторые участники ПЭК обозначаются специальными символами:

- символом B обозначается банк, и
- символом T обозначается продавец.

11.2 Примеры ПЭК

1. $A \to B : \xi_1(a_1), \ldots, \xi_k(a_k)$, где
   - $a_1, \ldots, a_k$ – переводы на одну и ту же сумму,
   - $\xi_1, \ldots, \xi_k$ – маскировка, которая коммутирует с ЭП S_B
2. B выбирает $i \in \{1, \ldots, k\}$, снимает маскировку у всех переводов кроме i-го, проверяет вскрытые переводы, после чего
   - $B \to A : S_B(\xi_i(a_i))$
   - B списывает со счёта A соотв. сумму.

   в результате B не может определить, что $a_i \in A$
3. $A \to T : S_B(a_i)$
4. $T \to B : S_B(a_i)$ (депонирование)
5. $B \to T$: деньги.

Маскировку ξ и ЭП S_B, коммутирующую с маскировкой, можно реализовать, например, следующим образом. Выбираются

- секретные простые числа p, q, и
- ХФ $h : \mathcal{A} \to \mathbf{Z}_n$ где $\mathcal{A}$ – множество всех переводов, и $n := pq$ (число n открыто).

Полагаем $K_B^+ := e$ и $K_B^- := d$, где $e, d \in \mathbf{Z}_n$, $ed \underset{\varphi(n)}{=} 1$. Ниже все вычисления - в $\mathbf{Z}_n$.

- $\forall a \in \mathcal{A} \quad \xi(a) := h(a) \cdot x^e$, где $x \underset{r}{\in} \mathbf{Z}_n \setminus \{0\}$
- $S_B(\xi(a)) := (h(a) \cdot x^e)^d = h(a)^d \cdot x$
- $S_B(a) = (a, h(a)^d)$
- Проверка ЭП (a, s) : $\quad h(a) \underset{n}{=} s^e \quad ?$

В некоторых модификациях данного КП числа e и d не постоянны, а зависят от денежного номинала $s(a)$, соответствующего переводу a. Например, число e может представлять собой произведение первых нечётных простых чисел, соответствующих единицам в двоичном представлении $s(a)$.

Недостаток: можно сделать копию перевода и использовать её. Чтобы этого избежать, надо заменить первое действие на

$$A \to B : \xi_1(a_1, N_1), \ldots, \xi_k(a_k, N_k)$$

Во 2-м действии B проверяет уникальность всех нонсов у демаскированных переводов. Когда в 4-м действии B получает $S_B(a_i, N_i)$, он проверяет, депонировался ли перевод с этим нонсом. Тем не менее, этот КП не может установить личность мошенника.

Можно модифицировать действие 4 следующим образом:

$$T \to B : S_B(a_i, N_i), N'$$

где нонс N' генерируется совместно A и T. Когда B получает это сообщение, он проверяет наличие N_i и N' в своей БД:

- если N' есть в БД банка, то присланное сообщение является не законным переводом, а копией, которую сделал T
- если N_i есть, а N' нет, то присланное сообщение является не законным переводом, а копией, которую сделал A

11.3 ПЭК, распознающий мошенника

1. $A \to B : \{\xi_i(a_i, N_i, [I_i^L], [I_i^R]) \mid i = 1, \ldots, k\}$, где
   - $a_1, \ldots, a_k$ – переводы на одну и ту же сумму,
   - $\xi_1, \ldots, \xi_k$ – маскировка, которая коммутирует с ЭП S_B
   - I_i^L и I_i^R - 2 доли секрета, где секрет $= \mathrm{Id}_A = I_i^L \oplus I_i^R$
2. B выбирает $i \in \{1, \ldots, k\}$, снимает маскировку у всех переводов кроме i-го, проверяет вскрытые переводы (в том числе, вскрывает $[I_i^L]$ и $[I_i^R]$), после чего
   - $B \to A : S_B(\xi_i(a_i, N_i, [I_i^L], [I_i^R]))$
   - B списывает со счёта A соотв. сумму.
3. $A \to T : S_B(a_i, N_i, [I_i^L], [I_i^R])$
4. $T \to A : \beta := (b_1, \ldots, b_k) \underset{r}{\in} \{L, R\}^k$
5. $A \to T : (I_1^{b_1}, \ldots, I_k^{b_k})$
6. $T \to B : \begin{cases} S_B(a_i, N_i, [I_i^L], [I_i^R]) \\ I := (I_1^{b_1}, \ldots, I_k^{b_k}) \end{cases}$ (депонирование)

7. B проверяет: $N_i \in \mathcal{N}$?

   - Если $N_i \notin \mathcal{N}$, то
     - $\mathcal{N} := \mathcal{N} \cup \{N_i\}$
     - $B \to T$: деньги
   - Если $N_i \in \mathcal{N}$, то B сравнивает I в ЭДП и в своей БД.
     - Если они совпадают, то перевод скопирован T.
     - Если они не совпадают, то перевод скопирован A, и т.к. он уже один раз использовался при покупке у другого продавца T', который выдал A строку $\beta' \neq \beta$, то $\exists j$: A послал T и T' разные половины I_j. B получает Id_A как $\oplus$ этих половин.

11.4 Электронные банкноты

Пусть участник A имеет ОК K_A^+ и ЗК $K_A^- =: x \supseteq \mathrm{Id}_A$.

ЭБ участника A представляет собой набор следующих данных:

$$\begin{cases} y := K_A^+ \\ r := g^k \ \% \ p, \text{ где } k \underset{r}{\in} \mathbf{Z}_p \\ S_B(\xi(\cdot)) \end{cases}$$

Платёж с использованием ЭБ осуществляется следующим образом.

1. $A \to T$: ЭМ
2. $T \to A$: $e \underset{r}{\in} \mathbf{Z}_p$
3. $A \to T$: $s := (k + ex) \ \% \ p$
4. T проверяет правильность s, используя y и r из ЭМ.

Депонирование:

$$T \to B : \ (\text{ЭМ}, e, s)$$

B проверяет законность использования этой ЭМ путём поиска в своей БД пары $(e', s') \neq (e, s)$ с теми же значениями y и r, что и у (e, s) (см. проверочное соотношение в ПАШ). Если такая пара нашлась, то B вычисляет x, по которому он определяет жулика.

Глава 12

Другие КП

12.1 Oblivious transfer

$A \to B : \frac{1}{2}\{d_1, d_2\}$.

1. $A \to B : K_1^+, K_2^+$
2. $B \to A : K_i^+(K)$, где $i \underset{r}{\in} \{1, 2\}$
3. $A \to B : \begin{cases} K_1^-(K_i^+(K))(d_1) \\ K_2^-(K_i^+(K))(d_2) \end{cases}$
4. $A \to B : K_1^-, K_2^-$ (для проверки честности)

КП Рабина $\frac{1}{2}$–раскрытия ЗК $= \{p, q\}$:

1. $A \to B : n := pq$
2. $B \to A : a := x^2 \ \% \ n$, где $x \underset{r}{\in} \mathbf{Z}_n^*$.
3. $A \to B : i \underset{r}{\in} \{x, n-x, y, n-y\} = \sqrt{a}$
   (A может вычислить $\sqrt{a}$, т.к. он знает p и q)
4. если B получил
   - y или $n-y$, то он может вычислить НОД$(x+y, n)$, который равен p или q
   - x или $n-x$, то B не может вычислить p и q

Дырка: B может найти такое a, что при известном $\sqrt{a}$ он всегда сможет найти p и q.

12.2 Заказное письмо

A хочет послать B сообщение так, чтобы B смог прочитать его только после того, как он распишется в получении этого сообщения. B хочет расписаться только после того, как установит, что он сможет прочесть сообщение.

1. $A \to B : K(m)$
2. A генерирует n пар ключей
$$\{(KA_i^L, KA_i^R := KA_i^L \oplus K) \mid i = 1, \ldots, n\}$$
3. $A \to B : \{KA_i^L(0), KA_i^R(0) \mid i = 1, \ldots, n\}$
4. B генерирует n пар ключей
$$\{(KB_i^L, KB_i^R) \mid i = 1, \ldots, n\}$$
и n пар сообщений
$$\{(MB_i^L, MB_i^R) \mid i = 1, \ldots, n\}$$
где каждая пара представляет собой расписку в получении сообщения m
5. $B \to A : \{\begin{pmatrix} KB_i^L(MB_i^L) \\ KB_i^R(MB_i^R) \end{pmatrix} \mid i = 1, \ldots, n\}$
6. $\forall i = 1, \ldots, n \quad A \to B : \frac{1}{2}\{KA_i^L, KA_i^R\}$
7. $\forall i = 1, \ldots, n \quad B \to A : \frac{1}{2}\{KB_i^L, KB_i^R\}$
8. A и B пытаются расшифровать компоненты полученных пар, в каждой паре будет расшифрована одна компонента
9. $\forall i = 1, \ldots, \quad \forall j = 1, \ldots, n$
   - $A \to B$: i-й бит KA_j^L, i-й бит KA_j^R
   - $B \to A$: i-й бит KB_j^L, i-й бит KB_j^R

   (биты пересылаются в зашифрованном виде)
10. A и B расшифровывают оставшиеся половины полученных пар
11. A и B обмениваются своими ЗК (для проверки честности)

На базе этого КП можно построить КП **одновременного обмена сообщениями**, решающего следующую задачу: участники A и B должны обменяться сообщениями, так, чтобы была исключена такая ситуация, при которой один из участников получил сообщение от партнёра, а своё не послал.

12.3 КП подписания контракта

КП подписания контракта (ППК) используется в тех случаях, когда участники A и B, которые не доверяют друг другу, хотят совместно подписать некоторое сообщение (называемое **контрактом**). Простое решение этой задачи, заключающееся в том, что

- один из участников подписывает контракт, после чего
- подписанный контракт пересылается другому участнику, который тоже подписывает этот контракт

не устраивает обоих участников, так как они не исключают возможности того, что партнёр может не подписать полученный контракт.

12.3.1 ППК с доверенным посредником

1. $A \to T:\ S_A(m)$ (m – это контракт)
2. $B \to T:\ S_B(m)$
3. $T \to A$: извещение, что он имеет $S_B(m)$
4. $T \to B$: извещение, что он имеет $S_A(m)$
5. $A \to B : S_A(m), S_A(m)$
6. $B \to A : S_B(S_A(m))$, вторая подписанная копия остаётся у B
7. $A \to T$: извещение, что он имеет контракт, подписанный обоими участниками
8. $B \to T$: извещение, что он имеет контракт, подписанный обоими участниками
9. T: уничтожает обе имеющиеся у него копии.

12.3.2 ППК без посредника

В данном КП используется симметричная ШС.

1. A генерирует n пар ключей

   $$\{(KA_i^L, KA_i^R) \mid i = 1, \ldots, n\}$$

   и n пар сообщений вида

   $$\{(MA_i^L, MA_i^R) \mid i = 1, \ldots, n\}$$

   где каждое сообщение содержит $S_A(m)$

2. B генерирует n пар ключей

$$\{(KB_i^L, KB_i^R) \mid i = 1, \ldots, n\}$$

и n пар сообщений вида

$$\{(MB_i^L, MB_i^R) \mid i = 1, \ldots, n\}$$

где каждое сообщение содержит $S_B(m)$

3. $A \to B : \{ \begin{pmatrix} KA_i^L(MA_i^L) \\ KA_i^R(MA_i^R) \end{pmatrix} \mid i = 1, \ldots, n\}$

4. $B \to A : \{ \begin{pmatrix} KB_i^L(MB_i^L) \\ KB_i^R(MB_i^R) \end{pmatrix} \mid i = 1, \ldots, n\}$

5. $\forall i = 1, \ldots, n \quad A \to B : \frac{1}{2}\{KA_i^L, KA_i^R\}$

6. $\forall i = 1, \ldots, n \quad B \to A : \frac{1}{2}\{KB_i^L, KB_i^R\}$

7. A и B пытаются расшифровать компоненты пар, в каждой паре будет расшифрована одна компонента

8. $\forall i = 1, \ldots, \quad \forall j = 1, \ldots, n$

   - $A \to B$: i-й бит KA_j^L, i-й бит KA_j^R
   - $B \to A$: i-й бит KB_j^L, i-й бит KB_j^R

9. A и B расшифровывают оставшиеся половины каждой пары

10. A и B обмениваются теми ЗК, которые использовались в ПЗ (для проверки честности)

Контракт считается подписанным, если

$$\begin{cases} \exists i : \ A \text{ имеет } (MB_i^L, MB_i^R) \\ \exists j : \ B \text{ имеет } (MA_j^L, MA_j^R) \end{cases}$$

12.4 КП ограниченной передачи секретов

КП, предназначенные для ограниченной передачи секретов, обозначаются знакосочетанием ANDOS (All-Or-Nothing Disclosure of Secrets).

12.4.1 Задача ограниченной передачи секретов

Пусть участник A имеет некоторое множество секретных сообщений

$$\{s_1, \ldots, s_k\} \subseteq \{0,1\}^n \tag{12.1}$$

Участники $B_1, B_2, \ldots$ хотят получить от A некоторые из этих секретов. Предполагается, что каждый из этих участников

- имеет право получить только один секрет из множества (12.1), и
- не сообщает A номер того секрета, который он хочет получить.

12.4.2 КП ограниченной передачи секретов честным участникам

Если все участники $B_1, \ldots$ честны, то для решения задачи, описанной в предыдущем пункте, можно использовать следующий КП.

Пусть B_1 хочет получить секрет s_{i_1}, B_2 хочет получить s_{i_2}, и т.д.

Выберем ШС RSA с таким модулем n, чтобы секреты из (12.1) можно было представить элементами множества $\mathbf{Z}_n$. Пусть e и d – ОК и ЗК этой ШС. Ниже все вычисления - в $\mathbf{Z}_n$. Участник B_j получает свой секрет s_{i_j} следующим образом.

1. $A \to B_j : \{b_i := s_i^e \mid i = 1, \ldots, k\}$
2. $B_j \to A : b := b_{i_j} \cdot x^e \quad (x \underset{r}{\in} \mathbf{Z}_n^*)$
3. $A \to B_j : c := b^d$
4. B вычисляет $s_{i_j} := c \cdot x^{-1}$

Если участники $B_1, \ldots$ нечестны и могут обмениваться друг с другом информацией, которую они получают от A, то они могут узнать больше секретов, чем им положено. Поэтому для ограниченной передачи секретов нечестным участникам нужно использовать другие КП.

12.4.3 КП ограниченной передачи секретов двум участникам

В этом пункте мы рассмотрим случай, когда число участников, желающих получить секреты, равно двум.

Выберем такую ШС, в которой |ОТ| = |ШТ|.

Ниже для каждой пары строк $x, y \in \{0,1\}^n$ знакосочетание $FBI(x,y)$ обозначает множество

$$\{i \geq 1 \mid x[i] = y[i]\ \}$$

1. A генерирует 4 ключа

$$K_1^+, \quad K_1^-, \quad K_2^+, \quad K_2^-$$

2. $A \to B_1 : K_1^+$

3. $A \to B_2 : K_2^+$

4. $B_1 \to B_2 : \beta_1, \ldots, \beta_k$, где $\forall i = 1, \ldots, k$

$$\beta_i \underset{r}{\in} \{0,1\}^n$$

5. $B_2 \to B_1 : \gamma_1, \ldots, \gamma_k$, где $\forall i = 1, \ldots, k$

$$\gamma_i \underset{r}{\in} \{0,1\}^n$$

6. $B_1 \to B_2 : P := FBI(\gamma_{i_1}, K_1^+(\gamma_{i_1}))$
(где i_1 – номер секрета, который хочет B_1)

7. $B_2 \to B_1 : Q := FBI(\beta_{i_2}, K_2^+(\beta_{i_2}))$
(где i_2 – номер секрета, который хочет B_2)

8. $B_1 \to A : \beta'_1, \ldots, \beta'_k$, где $\forall i = 1, \ldots, k, \ \forall j = 1, \ldots, n$

$$\beta'_i[j] := \begin{cases} \beta_i[j], & \text{если } j \in Q \\ 1 - \beta_i[j], & \text{иначе} \end{cases}$$

(из этого определения следует, что $\beta'_{i_2} = K_2^+(\beta_{i_2})$)

9. $B_2 \to A : \gamma'_1, \ldots, \gamma'_k$, где $\forall i = 1, \ldots, k, \ \forall j = 1, \ldots, n$

$$\gamma'_i[j] := \begin{cases} \gamma_i[j], & \text{если } j \in P \\ 1 - \gamma_i[j], & \text{иначе} \end{cases}$$

(из этого определения следует, что $\gamma'_{i_1} = K_1^+(\gamma_{i_1})$)

10. $A \to B_1 : \begin{cases} s_1 \oplus \gamma''_1, & \text{где } \gamma''_1 = K_1^-(\gamma'_1) \\ \ldots & \ldots \\ s_k \oplus \gamma''_k, & \text{где } \gamma''_k = K_1^-(\gamma'_k) \end{cases}$

(заметим, что $\gamma''_{i_1} = \gamma_{i_1}$)

11. $A \to B_2 : \begin{cases} s_1 \oplus \beta''_1, & \text{где } \beta''_1 = K_2^-(\beta'_1) \\ \dots & \dots \\ s_k \oplus \beta''_k, & \text{где } \beta''_k = K_2^-(\beta'_k) \end{cases}$
(заметим, что $\beta''_{i_2} = \beta_{i_2}$)

12. B_1 вычисляет $s_{i_1} := \gamma_{i_1} \oplus (s_{i_1} \oplus \gamma''_{i_1})$

13. B_2 вычисляет $s_{i_2} := \beta_{i_2} \oplus (s_{i_2} \oplus \beta''_{i_2})$

Дырки:

- A и B_2, действуя совместно, могут вычислить s_{i_1}: зная P, K_1^+, K_1^-, они могут подобрать i_1, и т.д.
- B_1 и B_2, действуя совместно, могут получить все секреты A

12.4.4 КП ограниченной передачи секретов нескольким участникам

Вышеприведённый КП можно обобщить до КП ограниченной передачи секретов нескольким участникам. Например, в случае трёх участников данный КП имеет следующий вид:

1. A генерирует 6 ключей
$$K_1^+, \quad K_1^-, \quad K_2^+, \quad K_2^-, \quad K_3^+, \quad K_3^-$$
2. $A \to B_1 : \ K_2^+, K_3^+$
3. $A \to B_2 : \ K_1^+, K_3^+$
4. $A \to B_3 : \ K_1^+, K_2^+$
5. и т.д.

12.5 Анонимная передача

Участники расположены по кругу. Каждый участник в каждый такт времени каждый участник подбрасывает монету с соседями следа и справа, и объявляет результат

Если A хочет передать сообщение, то он говорит противоположный результат в тех раундах, которые соответствуют 1 (единице) в битовом представлении его сообщения.

Если A видит, что полный выход КП не совпадает с сообщением, которое он пытался послать, он приходит к выводу, что кто-то другой тоже пытается послать сообщение, в этом случае он прекращает передачу, и ждёт некоторое время.

Сообщение можно шифровать ОК того участника, которому оно предназначено.