20 Протокол Диффи — Хеллмана

20.1 Головоломки Меркля

Пусть Алиса и Боб — абоненты некоторой информационной системы. Данные в системе передаются по открытым каналам связи, за которыми наблюдает злоумышленник Виктор. Предполагается, что Виктор не может изменять передаваемые данные. Для защиты от Виктора Алиса и Боб используют симметричную (блочную или поточную) криптосистему и выполняют шифрование сообщений на секретном ключе, известном только им двоим. Если в системе имеется n абонентов, то для организации взаимодействия всех возможных пар потребуется $n(n-1)/2$ ключей. Ключи следует распространять по секретным каналам связи, содержание которых обходится намного дороже, чем открытых. Возникает вопрос: можно ли обойтись без секретных каналов?

«Да», — ответил в 1972 году американский исследователь Р. Меркль. Меркль предложил следующий *протокол* (последовательность действий двух и более сторон, направленных на решение определенной задачи):

1. Алиса составляет и отправляет Бобу список из N головоломок, на решение каждой из которых требуется M секунд. Решение i-й головоломки содержит ключ K_i для связи с Бобом.

   В качестве головоломки можно использовать результат зашифрования пары (“`головоломка`”, K_i) на ключе K_i малой длины. Решение такой головоломки состоит в проведении атаки «грубой силой» по определению K_i при известном открытом тексте “`головоломка`”.

2. Боб выбирает головоломку со случайным номером i, решает ее за M секунд, определяет ключ K_i и отправляет Алисе сообщение “`Привет`”, зашифрованное на K_i.

3. Алиса просматривает ключи $K_1, \ldots, K_N$ и находит среди них ключ K_i, на котором было зашифровано сообщение “`Привет`”. Теперь Алиса и Боб располагают общим секретным ключом K_i.

Обсудим надежность протокола. Для определения K_i Виктор должен просмотреть ключи $K_1, \ldots, K_N$, а для этого решить не одну, а N головоломок и затратить не M, а NM секунд. Управляя N и M, Алиса может добиться того, что время NM окажется неприемлимо большим.

участники	время
Алиса	$O(N)$
Боб	$O(M)$
Виктор	$O(NM)$

20.2 Протокол Диффи — Хеллмана

В 1976 г. У. Диффи и М. Хеллман опубликовали работу «Новые направления в криптографии», в которой развили идеи Меркля. Авторы предложили следующий протокол:

Протокол Диффи — Хеллмана

Предназначен для выработки сторонами A и B общего ключа K

Стороны: A (Алиса), B (Боб).

Каналы: аутентифицируемый канал связи (АКС). При передаче данных по АКС обеспечивается контроль их целостности и подлинности.

Алгебраические структуры. Стороны используют вычисления в циклической группе $G = \langle g \rangle$ порядка q. Вырабатываемый ключ K является элементом G.

Шаги:

1. A: $a \xleftarrow{R} \{1, 2, \ldots, q-1\}$, вычислить g^a.

2. B: $b \xleftarrow{R} \{1, 2, \ldots, q-1\}$, вычислить g^b.

3. $A \to B$: g^a.

4. $A \leftarrow B$: g^b.

5. A: $K \leftarrow (g^b)^a$.

6. B: $K \leftarrow (g^a)^b$.

Корректность. Вырабатываемые сторонами ключи будут совпадать: $(g^b)^a = g^{ab} = (g^a)^b$.

Числа a и b называются *личными* ключами. Как и секретные ключи блочных и поточных криптосистем, личные ключи должны вырабатываться случайно, храниться в секрете и т.д.

Элементы g^a, g^b называются *открытыми* ключами. Личный ключ a однозначно определяет открытый ключ g^a и, наоборот, g^a однозначно определяет a. Открытые ключи обращаются в информационной системе в открытом виде и доступны в том числе и злоумышленнику Виктору. Однако, при надлежащем выборе g возведение в степень $a \mapsto g^a$ является вычислительно простой операцией, которую Алиса выполняет за приемлемое время, а дискретное логарифмирование $g^a \mapsto a$ — вычислительно трудной, с которой Виктор не справляется. В следующих лекциях мы проведем строгие рассуждения относительно вычислительной сложности и покажем, что времена работы участников для определенных групп G имеют следующий вид:

участники	время
Алиса	$(\log \lvert G\rvert)^{O(1)}$
Боб	$(\log \lvert G\rvert)^{O(1)}$
Виктор	$e^{O(\log \lvert G\rvert)}$ (экспоненциальное) или $e^{o(\log \lvert G\rvert)}$ (субэкспоненциальное)

В таблице мы использовали символы Ландау o и O. Напомним, что

– запись $g(n) = o(f(n))$ означает, что $g(n)/f(n) \to 0$ при $n \to \infty$;

– запись $g(n) = O(f(n))$ означает, что $|g(n)| \leqslant Cf(n)$ для некоторой константы C при достаточно больших n.

Для атаки на протокол Виктору не обязательно находить личные ключи сторон. Достаточно по известным g^a и g^b определить ключ K, т. е. g^{ab}. Данная задача, известная как *вычислительная задача Диффи — Хеллмана* (CDH, Computational Diffie — Hellman), также признается вычислительно трудной.

Отметим, что алгоритм решения задачи дискретного логарифмирования может быть трансформирован в алгоритм решения CDH, однако, справедливость обратного утверждения является на сегодняшний день одной из нерешенных проблем теоретической криптографии.

Протокол Диффи — Хеллмана может быть преобразован в протокол выработки общего ключа тремя и более абонентами. Действительно, пусть имеется третий абонент C (Клара). Клара генерирует $c \leftarrow \{1, 2, \ldots, q-1\}$, находит g^c и все абоненты обмениваются между собой открытыми ключами.

Далее стороны выполняют следующие пересылки:

$$A \to B\colon (g^c)^a$$
$$B \to C\colon (g^a)^b$$
$$C \to A\colon (g^b)^a.$$

По окончании пересылок стороны находят общий ключ:

$$A\colon \left((g^b)^c\right)^a$$
$$B\colon ((g^c)^a)^b$$
$$C\colon \left((g^a)^b\right)^c.$$

Упражнение 20.1. Разработать протокол для выработки общего ключа n абонентами. □

20.3 Атака «противник посередине»

Канал АКС занимает промежуточное положение между ОКС и СКС. Реализовать АКС проще (и дешевле), чем СКС. Представьте, что Боб уехал в далекую страну, все каналы связи с ним проходят по дну океана в специальном кабель-канале и потенциально прослушиваются Виктором. Алиса вполне резонно сомневается в конфиденциальности пересылаемых Бобу данных. Алиса могла бы зашифровать данные, но для этого нужен ключ, который снова требуется передать конфиденциально! С организацией СКС ничего не получается. С другой стороны, Алиса и Боб вполне могут организовать АКС. Например, Алиса и Боб могут обменяться данными по электронной почте (ОКС), а затем проверить целостность и подлинность данных по телефону (АКС). Контроль подлинности данных поддерживается голосовой (характерный тембр голоса) и вербальной (характерные жаргонизмы) аутентификацией абонентов, целостность — проверкой контрольных характеристик переданных данных. Отметим, что телефонный АКС рекомендован в системе PGP (защищенная электронная почта) при обмене открытыми ключами.

При выполнении протокола Диффи — Хеллмана стороны вырабатывают общий секрет K, с помощью которого могут организовать СКС. Таким образом, протокола Диффи — Хеллмана фактически позволяет преобразовать АКС в СКС.

Можно ли отказаться от АКС и передавать открытые ключи по ОКС? Оказывается, что нет.

Если CDH — вычислительно трудная задача, то протокол Диффи — Хеллмана является стойким относительно атак пассивного злоумышленника Виктора. Однако, ситуация кардинальным образом меняется, если у Виктора есть возможность не только перехватывать, но и менять пакеты, циркулирующие по открытым каналам связи.

Виктор может провести следующую атаку, которая называется «противник посередине» (MITM, man-in-the-middle в англоязычной литературе):

1. Виктор вырабатывает случайные a', b' и меняет g^a на $g^{a'}$, а g^b на $g^{b'}$.
2. По окончании пересылок сторона A сформирует ключ $K_1 = (g^{b'})^a$, а сторона B — ключ $K_2 = (g^{a'})^b$.
3. Оба ключа известны Виктору: $K_1 = (g^a)^{b'}$, $K_2 = (g^b)^{a'}$!

Пример 20.1 (CAPTCHA). Атака «противник посередине» может применяться не только для криптографических протоколов. Рассмотрим один интересный пример, связанный с тестами CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart). В этих тестах машина (компьютер) проверяет, что взаимодействует с человеком. Машина преобразует текст в графическую картинку, искажая начертания символов, и предлагает восстановить по картинке текст. Человек это делает легко, а вот машина (другой компьютер) — нет.

Известен случай, когда для автоматического прохождения тестов с сайта A хакеры разработали сайт B, на котором эти тесты дублируют. Тесты проходят люди, их ответы персылаются на сайт A и засчитываются как правильные. □

20.4 Реализация протокола Диффи — Хеллмана

Для реализации протокола Диффи — Хеллмана, а также других криптографических систем с открытыми ключами, нам требуется построить циклическую группу G порядка q. (Выбор конкретных групп, задание конкретных базовых криптографических алгоритмов и т. д. называется *инстанциированием* протокола.)

Упражнение 20.2. Доказать, что если q — простое, то G — циклическая. □

Существуют разные способы построения G, мы рассмотрим способ, состоящий во вложении G в подходящую мультипликативную группу $\mathbb{F}_p^*$. Пусть p — простое, пусть известна факторизация $p-1$ и пусть $q \mid p-1$.

В первом семестре мы доказали, что $\mathbb{F}_p^*$ — циклическая группа. Примтивный элемент, который порождает данную группу, может быть найден с помощью следующего алгоритма.

Алгоритм Генерация примитивного элемента

Вход: простое p и факторизация $p-1 = p_1^{e_1} \dots p_k^{e_k}$, p_i — простые.

Выход: примитивный элемент $\alpha \in \mathbb{F}_p^*$.

Шаги:

1. $\alpha \xleftarrow{R} \mathbb{F}_p^*$.
2. Для $i = 1, \ldots, k$:
   (1) если $\alpha^{(p-1)/p_i} = 1$, то перейти к шагу 1.
3. Вернуть α.

Корректность. Предположим, что в результате работы алгоритма получен элемент α, который не является примитивным. По теореме Лагранжа $\operatorname{ord}\alpha \mid\neq \prod p_i^{e_i}$ и, следовательно, найдется p_i такой, что $\operatorname{ord}\alpha \mid \frac{p-1}{p_i}$. Но тогда $\alpha^{(p-1)/p_i} = 1$, что противоречит результату проверки на шаге 2(1).

Пример 20.2. Элемент $2 \in \mathbb{F}_{23}^*$ не является примитивным:

$$2^{11} = 2^5 \cdot 2^5 \cdot 2 \equiv 9 \cdot 9 \cdot 2 = 1 \pmod{23}.$$

Элемент $5 \in \mathbb{F}_{23}^*$ является примитивным:

$$5^2 \equiv 2 \pmod{23}, \quad 5^{11} = 5(5^2)^5 \equiv 5 \cdot 2^5 \equiv -1 \pmod{23}.$$

После нахождения α элемент g, который имеет заданный порядок $q \mid p-1$, можно определить с помощью следующего алгоритма. Элемент g порождает циклическую подгруппу $G \subseteq \mathbb{F}_p^*$ нужного порядка q.

Алгоритм Генерация элемента заданного порядка

Вход: простое p, q — делитель $p-1$, α — примитивный элемент $\bmod p$.
Выход: $g \in \mathbb{F}_p^*$ — элемент порядка q.
Шаги:

1. $g \leftarrow \alpha^{(p-1)/q} \bmod p$.
2. Вернуть g.

Корректность. Покажем, что $\operatorname{ord} g = q$. Действительно,

(i) $g^q = \alpha^{p-1} = 1 \Rightarrow \operatorname{ord} g \leqslant q$;

(ii) если $\operatorname{ord} g < q$, то $1 = g^{\operatorname{ord} g} = \alpha^{(p-1)\operatorname{ord} g/q} \Rightarrow \operatorname{ord}\alpha \leqslant (p-1)\operatorname{ord} g/q < p-1$ что противоречит примитивности α.

Если q — простое, то найти g порядка q можно без предварительного определения примитивного элемента.

Алгоритм Генерация элемента простого порядка

Вход: простые p и q, $q \mid p-1$.
Вход: g — элемент порядка q по модулю p.
Шаги:

1. $\alpha \xleftarrow{R} \mathbb{F}_p^*$.
2. $g \leftarrow \alpha^{(p-1)/q} \bmod p$.
3. Если $g = 1$, то перейти к шагу 1.
4. Возвратить g.

В новом алгоритме выходное число g удовлетворяет условиям: $g \neq 1$, $g^q = \alpha^{p-1} \overset{\text{м.т.Ф.}}{\equiv} 1 \pmod{p}$. Поэтому $\operatorname{ord} g \neq 1$, $\operatorname{ord} g \mid q$ и, в силу простоты q, $\operatorname{ord} g = q$.

Упражнение 20.3. Привести пример реализации протокола Диффи — Хеллмана при $q = 11$ и $p = 23$ (найти g, выработать личные и открытые ключи, определить общий ключ K). □

20.5 Анализ алгоритмов (забегая вперед)

Арифметика больших чисел. Пусть мы выполняем мультипликативные операции над l-битовыми числами.

Аддитивные операции: $O(l)$.

Мультипликативные операции: $O(l^2)$.

Возведение в l-битовую степень: $O(l^3)$.

Генерация примитивного элемента. Алгоритм работает без ошибок. Вероятность успеха алгоритма за 1 проход есть $\beta = \varphi(p-1)/(p-1)$, где $\varphi(p-1)$ — количество примтивных элементов по модулю p. Для любого натурального $n \geqslant 3$ справедлива оценка

$$\frac{\varphi(n)}{n} > \frac{\ln 2}{2 \ln n}.$$

Поэтому $\beta = \Omega(1/\log p)$ и среднее число проходов:

$$\sum_{t=1}^{\infty} t\mathsf{P}\left\{\text{нужно } t \text{ проходов}\right\} = \sum_{t=0}^{\infty} \mathsf{P}\left\{\text{нужно } > t \text{ проходов}\right\} = \sum_{t=0}^{\infty} (1-\beta)^t = \frac{1}{1-(1-\beta)} = \frac{1}{\beta} = O(\log p).$$

Трудоемкость прохода: $O(\log^4 p)$. Среднее время работы алгоритма: $O(\log^5 p)$.

Генерация элемента простого порядка. Самостоятельно.