4. ОТКРЫТОЕ РАСПРОСТРАНЕНИЕ КЛЮЧЕЙ

4.1. Теоретические сведения

Современные симметричные шифры обладают высокой практической криптографической стойкостью, обеспечивающей требуемую конфиденциальность каналов связи. Однако симметричная криптография сталкивается с существенными трудностями при решении двух задач.

1) Распределение секретного ключа. До начала обмена зашифрованными сообщениями обе стороны должны иметь одинаковый общий секретный ключ. Исторически эта задача решалась путем личной встречи или передачи ключа с надежным курьером. Однако в современных информационных системах такое решение практически не реализуемо.

2) Обеспечение защиты сообщения от подделки и подтверждение авторства сообщения (цифровая подпись). Не решается симметричными системами без наличия третьего доверенного лица.

Для решения этих задач в 1976 году Уитфилд Диффи (Whitfield Diffie), Мартин Хеллман (Martin Hellman) и Ральф Меркл (Ralph Merkle) предложили идею использования односторонних функций и односторонних функций с потайным ходом (с секретом). Эта идея положила начало асимметричной криптографии, в рамках которой проблемы симметричной криптографии оказываются элегантно разрешимыми.

Понятие односторонней функции является базовым в криптографии.

Односторонняя функция – это некоторая функция f, такая, что для любого x из ее области определения f(x) легко вычислима; однако практически для всех y из ее области значений нахождение x, для которого $y = f(x)$, вычислительно неосуществимо.

Однако такое определение оставляет открытым вопрос о том, что означает «вычислительно неосуществимо».

Поэтому более информативным является другое определение.

Пусть $\{0,1\}^n$ – множество всех двоичных строк длиной n.

Функция $f: \{0,1\}^* \to \{0,1\}^*$ является односторонней функцией, если она эффективно вычисляется за полиномиальное время на детерминированной машине Тьюринга, но не существует полиномиальной вероятностной машины Тьюринга, которая обращает эту функцию с более чем экспоненциально малой вероятностью. То есть для любой вероятностной полиномиальной машины M, для любого полинома $p(n)$ и достаточно большого $n \in N$ выполняется:

$$\Pr\left[M\left(f(m)\right) \in f^{-1}(m)\right] < 1/p(n),$$

где m – случайная равновероятная строка из множества $\{0,1\}^n$.

Время работы машины M ограничено полиномом от длины искомого прообраза. Время работы машины M определяется полиномом от длины искомого образа.

В настоящее время существование односторонних функций не доказано. Проблема заключается в следующем. Если f является односторонней функцией, то нахождение обратной функции является трудновычислимой, но легко проверяемой задачей. Тогда из существования односторонней функции следует, что $P \neq NP$. Однако неизвестно, следует ли из $P \neq NP$ существование односторонних функций.

В информационных системах нашли широкое применение односторонние функции, **сохраняющие длину**, – односторонние функции, битовая длина значения которых равна битовой длине аргумента. Такие функции используют для построения генераторов псевдослучайных последовательностей чисел. Для построения криптографических хеш-функций используют односторонние функции, длина значения которых постоянна при любой длине аргумента.

Построение систем шифрования на основе односторонних функций является достаточно трудной задачей, что хорошо видно из следующего примера. Пусть имеется односторонняя функция f. Необходимо построить криптосистему с открытым ключом. Криптосистема определяется семейством функций зашифрования E и семейством функций расшифрования D:

$$(\forall m)\ D\big(E(m)\big) = m,$$

где m – исходное сообщение.

Пусть для вычисления криптограммы c использовалась функция f, т. е. $c = f(m)$. Тогда противник, перехвативший зашифрованное сообщение c, может вычислить исходное сообщение m с пренебрежимо малой вероятностью. Однако и полномочный получатель столкнется с той же проблемой. Кроме того, из того что f является односторонней функцией, следует, что противник не сможет вычислить сообщение целиком, но не очевидно, что не сможет вычислить часть сообщения, что является существенной уязвимостью такой криптосистемы.

Кандидаты в односторонние функции. В настоящее время существует несколько десятков кандидатов в односторонние функции, наиболее широкое применение в криптографии нашли следующие:

– умножение и факторизация;

– возведение в квадрат и вычисление квадратного корня по модулю;

– возведение в степень по модулю и дискретное логарифмирование;

– криптографические хеш-функции.

Умножение и факторизация

Аргументами функции f являются пары взаимно простых чисел p и q, значение функции

$$N = f(p, q) = pq.$$

Значение функции может быть вычислено за время порядка $O(n^2)$, где n – сумма длин значений аргументов в битах.

Для нахождения значений обратной функции требуется разложить N на пару взаимно простых целых множителей.

Существует несколько методов разложения на простые множители, например:

– метод факторизации Ферма;

– метод эллиптической кривой;

– квадратичное решето;

– квадратичное решето в числовом поле.

Некоторые из них эффективны только для чисел специального вида. Верхняя оценка сложности метода Ферма и полного перебора порядка – $O(\sqrt{p})$.

Возможна факторизация с полиномиальной сложностью на квантовом компьютере методом Шнорра.

Возведение в квадрат и вычисление квадратного корня по модулю

Аргументами функции f являются пара простых чисел x и N, где $N = pq$, p и q – простые числа:

$$f(x) = x^2 \bmod N.$$

Для нахождения обратной функции требуется вычисление квадратного корня по модулю N, то есть нахождение x, если известно y при $x^2 \bmod N = y$. Эта задача имеет такую же сложность, как и разложение N на множители.

На основе функции возведения в квадрат и вычисления квадратного корня по модулю построена, а также задачи факторизации построена криптографическая схема Рабина.

Возведение в степень по модулю и дискретное логарифмирование

Параметрами функции f являются простое число p и целое число a, $0 < a < p$. Аргументом функции f является целое число $0 < x < p$:

$$f(x) = a^x \bmod p.$$

Функция f может быть вычислена за время $O(n^3)$, где $n = \log_2 p$.

Пусть $(G,*)$ – конечная абелева группа. Задача вычисления дискретных логарифмов заключается в нахождении целого числа x, удовлетворяющего соотношению $a^x \bmod p = B$, при известных a, B.

Сложность нахождения дискретного логарифма существенно зависит от вида группы $(G,*)$.

Дискретное логарифмирование аналогично обычному логарифмированию в поле действительных чисел. Однако, в отличие от последней задачи, в которой решение является приближенным, задача о вычислении дискретного логарифма имеет точное решение.

Криптографические хеш-функции

Примером односторонних криптографических хеш-функций является SHA-2 – семейство криптографических алгоритмов однонаправленных хеш-функций, включающее в себя алгоритмы SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/256 и SHA-512/224. Хеш-функции предназначены для создания образов фиксированной длины для сообщений произвольной длины. Применяются в различных приложениях или компонентах, связанных с защитой информации, таких, как аутентификация, контроль целостности, цифровая подпись.

Протокол Диффи – Хеллмана

Протокол Диффи – Хеллмана – это криптографический протокол, позволяющий двум и более сторонам сгенерировать общий секрет, используя незащищенный канал связи. Под незащищенным каналом понимают канал, который может быть наблюдаем (прослушан) третьей стороной. Общий секрет используют для генерации ключа, который затем применяют в симметричных алгоритмах шифрования (для зашифрования и расшифрования) и защищенного обмена сообщениями.

Пусть имеются два несекретных числа g и P, известные всем заинтересованным сторонам. P – большое простое число, g – является первообразным корнем по модулю P. Первообразным корнем по модулю P называют такое число g, что все его степени по модулю P принимают значения всех чисел, взаимно простых с P.

Пусть также имеются два абонента Алиса и Боб, которые поочередно выполняют следующие шаги:

1) Алиса генерирует целое число a и держит его в секрете, затем вычисляет $A = g^a \bmod P$ и пересылает его Бобу.

2) Боб генерирует целое число b и держит его в секрете, затем вычисляет $B = g^b \bmod P$ и пересылает его Алисе.

3) Алиса вычисляет значение $B^a \bmod P = g^{ab} \bmod P$.

4) Боб вычисляет значение $A^b \bmod P = g^{ab} \bmod P$.

Нетрудно видеть, что Алиса и Боб вычислили одно и то же число:

$$K = g^{ab} \bmod P = g^{ba} \bmod P.$$

Теперь Алиса и Боб могут сгенерировать ключ шифрования, используя общий секрет в качестве стартового значения генератора (одинакового у Алисы и Боба). В качестве такого генератора часто используют криптографические функции хеширования.

Использование протокола Диффи – Хеллмана не ограничивается двумя участниками. Он может быть применен на неограниченное количество пользователей. Например, в случае трех участников A, B и C они вычисляют общий секрет:

$$K = g^{abc} \bmod P = g^{cba} \bmod P = g^{bac} \bmod P.$$

При этом злоумышленник может наблюдать в открытом канале промежуточные значения $g^a, g^b, g^c, g^{ab}, g^{ac}, g^{bc}$, но не может вычислить общий секрет g^{abc}.

Возвращаясь снова к протоколу с двумя участниками, предполагается, что злоумышленник может получить значения A, B, g и P, но не модифицировать. Если числа A, B и P выбраны достаточно большими, то при попытке вычислить общий секрет атакующий встретится с задачей, неразрешимой за разумное время. Величина числа g на стойкость протокола не влияет, поэтому его обычно выбирают пределах первого десятка (из соображений простоты вычислений).

Величину P на практике выбирают не менее 1024 бит.

В 2016 году была представлена работа, показавшая возможности по подготовке специальных конечных полей для алгоритма Диффи – Хеллмана. Выбранное исследователями простое число p специального вида (размером 1024 бита) выглядит обычным для пользователей, но упрощает на несколько порядков сложность вычислений по методу SNFS (Специальный метод решета числового поля – Special Number Feld Sieve) для решения задачи дискретного логарифмирования. Для борьбы с атакой предлагается увеличить размер модуля до 2048 бит.

При правильном выборе параметров протокол Диффи – Хеллмана устойчив к пассивным атакам. Однако он не способен противостоять атаке «человек посредине». Пусть имеются два абонента Алиса – А и Боб – Б, а также атакующий Джо – Д. Джо может перехватывать и модифицировать сообщения. Последовательность действий участников при атаке «человек посредине» приведена в табл. 4.1.

Таким образом, Джо получает общий секрет для защищенного обмена сообщениями с Алисой и секрет для общения с Бобом, выдавая себя за Боба для Алисы и за Алису для Боба. Защитой от такой атаки являются протоколы аутентификации сторон.

Поскольку при выполнении шагов протокола показатели степенной функции достаточно велики, а также велико значение числа *P*, то для ускорения вычислений используют **алгоритм быстрого возведения в степень методом повторяющихся возведений в квадрат и умножения**.

Таблица 4.1

Атака «человек посредине»

№	Алиса	Джо	Боб
1	$g^a \bmod P \rightarrow$	$g^a \bmod P$	–
2	$g^j \bmod P \leftarrow$	$g^j \bmod P$	–
3	$g^{ja} \bmod P$	$g^{aj} \bmod P$	–
4	–	$g^j \bmod P \rightarrow$	$g^j \bmod P$
5	–	$g^b \bmod P \leftarrow$	$g^b \bmod P$
6	–	$g^{bj} \bmod P$	$g^{jb} \bmod P$

Метод заключается в следующем.

Пусть требуется вычислить $x^a \bmod n$.

Представим показатель степени в виде

$$a = a_{j-1}2^{j-1} + a_{j-2}2^{j-2} + \cdots + a_2 2^2 + a_1 2^1 + a_0,$$

где $a_j = (0,1)$.

Далее представим $x^a \bmod n$ в виде

$$x^a \bmod n = x^{a_{j-1}2^{j-1}+a_{j-2}2^{j-2}+\cdots+a_2 2^2 + a_1 2^1 + a_0} \bmod n =$$

$$= (x^2)^{a_{j-1}2^{j-2}+a_{j-2}2^{j-3}+a_1 2^0} x^{a_0} \bmod n =$$

$$= ((x^2)^2)^{a_{j-1}2^{j-3}+a_{j-2}2^{j-4}+\cdots+a_2 2^0} (x^2)^{a_1} x^{a_0} \bmod n =$$

$$= (\ldots((x^2)^2 \ldots)^2)^{a_{j-1}} \ldots (x^8)^{a_3} (x^4)^{a_2} (x^2)^{a_1} x^{a_0} \bmod n.$$

Затем вычисляют $x^2 \bmod n$ и выполняют замену в преобразованном выражении. Вычисление производят до тех пор, пока не будет получен результат.

4.2. Задание для самостоятельного выполнения

Для заданного простого P (в соответствии с вариантом) найти g – примитивный элемент конечного поля GF(P) и выполнить генерацию общего секрета. Для нахождения g воспользуйтесь методом перебора по возрастанию, возведения в степень по модулю P и проверки того факта, что все степени принимают значения от 0 до P – 1.

Варианты:

1) 5717	11) 3877	21) 4877
2) 9721	12) 1877	22) 2957
3) 2111	13) 1973	23) 2971
4) 3917	14) 4937	24) 3137
5) 4231	15) 7237	25) 1123
6) 9001	16) 9011	26) 9679
7) 8699	17) 8233	27) 8329
8) 8447	18) 8581	28) 7351
9) 7489	19) 7573	29) 7673
10) 7759	20) 7883	30) 6823

Вариант выбирается в соответствии с порядковым номером студента в рамках группы. Если студентов в группе больше, чем вариантов в списке, то варианты снова повторяются, начиная с единицы.

Отчет должен содержать:

1) Листинги программ:

а) для проверки g (первообразный корень по модулю P);

б) для вычисления $B^a \bmod P = g^{ab} \bmod P$ и $A^b \bmod P = g^{ab} \bmod P$.

2) Описание шагов, выполняемых участниками протокола – Алисой и Бобом для вычисления общего секрета.

3) Выводы, содержащие:

а) модель атакующего и оценки длины ключа;

б) возможные угрозы протоколу и предложения по защите от них.

Важно!

Возведение в степень выполнять методом последовательного возведения в квадрат и умножения.

При возведении в степень по модулю операцию взятия по модулю выполнять на каждом шаге возведения в квадрат и умножения для исключения переполнения.