ЧАСТЬ 2

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Асимметричные системы шифрования. Смысл *асимметричных криптосистем* (*системы открытого шифрования, с открытым ключом — public key systems*) состоит в том, что для зашифрования и расшифрования используются разные преобразования. Одно из них — зашифрование — является абсолютно открытым для всех. Другое же — расшифрование — остается секретным за счет секретности ключа расшифрования. Таким образом, любой, кто хочет что-либо зашифровать, пользуется открытым преобразованием, но расшифровать и прочитать это сможет лишь тот, кто владеет секретным ключом. Схема асимметричной криптосистемы представлена на рис. 2.1.

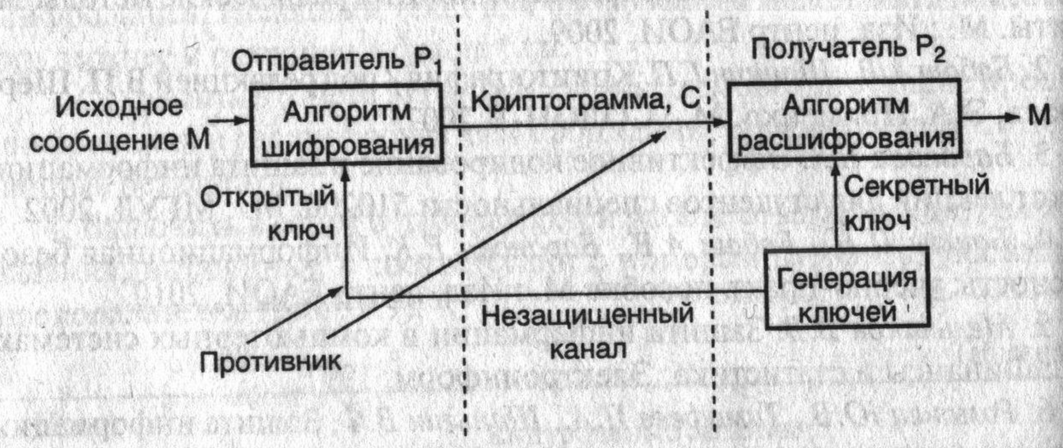

Рис. 2.1. Обобщенная схема асимметричной криптосистемы

В настоящее время во многих асимметричных криптосистемах вид преобразования определяется ключом. У пользователя есть два ключа — секретный и открытый. Открытый ключ публикуется в общедоступном месте, и каждый, кто хочет послать сообщение этому пользователю, зашифровывает текст открытым ключом. Расшифровать сообщение может только упомянутый пользователь с секретным ключом. Таким образом, отпадает проблема передачи секретного ключа, как в симметричных системах. Однако, несмотря на все свои преимущества, эти криптосистемы достаточно трудоемки и медлительны. Стойкость асимметричных криптосистем базируется в основном на алгоритмической трудности решить за приемлемое время какую-либо задачу. Если злоумышленнику удастся построить такой алгоритм, то дискредитирована будет вся система и все сообщения, зашифрованные с помощью этой системы. В этом состоит главная опасность асимметричных криптосистем в отличие от симметричных.

Алгоритм Диффи — Хеллмана. Алгоритм Диффи — Хеллмана (Diffie — Hellman) использует функцию дискретного возведения в степень. Вначале генерируются два больших простых числа n и q. Эти два числа не обязательно хранить в секрете. Далее один из партнеров P_1 генерирует случайное число x и посылает другому участнику будущих обменов P_2 значение

$$A = q^x \bmod n.$$

По получении значения A партнер P_2 генерирует случайное число y и посылает участнику обмена P_1 вычисленное значение

$$B = q^y \bmod n.$$

Партнер P_1, получив значение B, вычисляет $K_x = B^x \bmod n$, а партнер P_2 — $K_y = A^y \bmod n$. Алгоритм гарантирует, что числа K_y и K_x равны и могут быть использованы в качестве секретного ключа для шифрования. Даже перехватив числа A и B, трудно вычислить K_x или K_y. Схематично работа алгоритма Диффи — Хеллмана представлена на рис. 2.2.

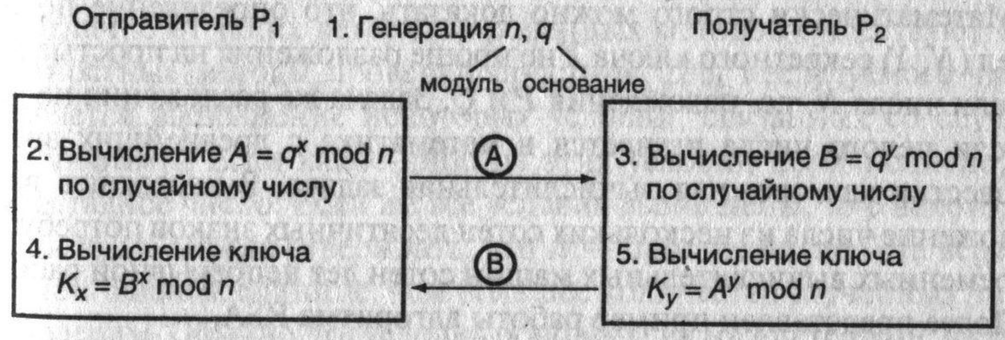

Рис. 2.2. Алгоритм Диффи — Хеллмана

Пример 2.1

$n = 5$, $q = 7$, $x = 3$, $y = 2$

$A = 7^3 \pmod 5 = 343 \pmod 5 = 3$; $B = 7^2 \pmod 5 = 49 \pmod 5 = 4$;

$K_x = 7^3 \pmod 5 = 64 \pmod 5 = 4$; $K_y = 3^2 \pmod 5 = 4$.

Алгоритм RSA. Первое практическое воплощение принцип открытого шифрования получил в системе RSA, разработанной в 1977 г.

в Массачусетском технологическом институте (США) и получившей свое название от первых букв фамилий авторов: Рональд Ривест (R. Rivest), Эди Шамир (A. Shamir), Леонард Адлеман (L. Adleman).

Идея авторов этого алгоритма состояла в том, что, взяв целое число N в виде произведения двух больших простых чисел $N = P \cdot Q$, легко подобрать пару чисел Y и X, таких, чтобы для любого целого числа M, меньшего N, было справедливо соотношение

$$(M^X)^Y = M \bmod N.$$

В качестве открытого ключа шифрования в системе RSA выступают ключ Y и модуль N, а секретным ключом для расшифрования сообщений является число X. Процедура шифрования сообщения M, рассматриваемого как целое число (такое допущение возможно вследствие того, что любой контент может быть представлен в числовой форме при обработке в средствах вычислительной техники), меньшее N (при необходимости длинное сообщение разбивается на отрезки, шифруемые независимо), состоит в вычислении значения

$$C = M^Y \bmod N.$$

Расшифрование осуществляется аналогично с использованием секретного ключа X:

$$M = C^X \bmod N.$$

Математически строго можно доказать, что определение по паре чисел (N, Y) секретного ключа X не проще разложения на простые множители числа N, т.е. нахождения P и Q. Задача же разложения на множители целого числа изучается в математике с древнейших времен и известна как сложная вычислительная задача. В настоящее время разложение числа из нескольких сотен десятичных знаков потребует от современных вычислительных машин сотен лет непрерывной работы.

Далее представлен пример работы алгоритма RSA.

Генерация ключей

Получатель 1. P, Q — простые, $N = P \cdot Q$

2. $\varphi(N) = (P-1) \cdot (Q-1)$, $\varphi(N)$ — функция Эйлера

Выбор открытого ключа Y:

$1 < Y \le \varphi(N)$, НОД(Y, $\phi(N)$) = 1

Выбор открытого ключа X:

$X \cdot Y \equiv 1 \pmod{\varphi(N)}$

Отправитель **шифрование** M ($M_i = 0, 1, 2, \ldots, N-1$)

3. $C_i = M_i^Y \pmod N$

Получатель **расшифрование** C (C_1, C_2, ..., C_i, ...)

4. $M_i = C_i^X \pmod N$

Пример

Генерация ключей

1. $P = 3$, $Q = 11$, $N = P \cdot Q = 33$
2. $\varphi(N) = (P-1) \cdot (Q-1)$, $\varphi(N) = 1$
   $Y = 7$, НОД(Y, $\varphi(N)$) = 1
   $X \cdot Y = 1 \pmod{20}$, $7 \cdot 3 = 1 \pmod{20}$, X = 3

Сообщение: $M_1M_2 \rightarrow 32$; $M_1 = 3 < 33$, $M_2 = 2 < 33$

Шифрование $C_i = M_i^Y \pmod N$

3. $C_1 = 3^7 \bmod 33 = 2187 \bmod 33 = 9$
   $C_2 = 2^7 \bmod 33 = 128 \bmod 33 = 29$

Расшифрование $M_i = C_i^X \pmod N$

4. $M_1 = 9^3 \bmod 33 = 729 \bmod 33 = 3$
   $M_2 = 29^3 \bmod 33 = 24389 \bmod 33 = 2$

Методы проверки чисел на простоту. Одна из главных проблем асимметричного шифрования — генерация больших простых чисел. Простейшим методом проверки простоты натурального числа N является метод пробных делений: для $d = 2, 3 \ldots$ проверяется выполнение условия (d, N) >1 (здесь (d, N) — наибольший общий делитель чисел d, N). Число операций, требуемых для этого метода, имеет порядок $\sqrt{N}$. Поэтому уже для чисел порядка 10^{30}—10^{40} этот метод не применим.

В отличие от таких детерминированных методов существуют еще вероятностные методы проверки простоты. Для исследуемого числа проверяется выполнение некоторых условий, связанных со случайными числами. Если какое-либо из этих условий не выполнено, то N — составное число. Если же все условия выполнены, то с некоторой вероятностью можно утверждать, что N — простое число. Эта вероятность тем ближе к единице, чем большее количество случайных чисел мы проверим. Обычно эти условия основаны на малой теореме Ферма, утверждающей, что для любого положительного числа b, не превосходящего некоторого простого числа p,

$$b^{(p-1)} = 1 \pmod p.$$

Например, $2^6 = 64 = 63 + 1 = 1 \pmod 7$. Если требуется определить, является ли целое число r простым, то можно выбрать любое положительное целое число b, меньшее r, и проверить, выполнено ли равенство

$$b^{(r-1)} = 1 \pmod r.$$

Если равенство не выполнено, то на основании теоремы Ферма можно быть совершенно уверенным, что r — не простое число. Если же равенство выполнено, то можно лишь предполагать, что r — простое число, и поэтому назвать его псевдопростым по основанию b. Вероятность $P(x)$ того, что составное число x окажется псевдопростым по случайному основанию, убывает с ростом x.

К сожалению, существуют так называемые числа Кармайкла — составные числа, которые обладают свойством

$b^{(r-1)} = 1 (\text{mod } r)$ для всех b из интервала $[1, r]$, которые взаимно просты с r.

Примером числа Кармайкла является число $561 = 3 \cdot 11 \cdot 17$.

Классический результат теории чисел — теорема Чебышева — показывает, что доля положительных целых чисел, меньших некоторого целого m и являющихся простыми, близка к $1/(\ln m)$. Например, доля целых чисел, меньших 10^{100} и являющихся простыми, близка к $1/(\ln 10^{100}) = 1/230$. Таким образом, если выбрать случайно большое целое положительное нечетное число x и последовательно проверять на простоту числа $x, x + 1, x + 2, \ldots$, то в среднем впервые простое число встретится на шаге с номером $\ln x$.