1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Концепция криптографии с открытым ключом была предложена Уитфилдом Диффи (Whitfield Diffie) и Мартином Хеллманом (Martin Hellman), и, независимо от них, Ральфом Мерклом (Ralph Merkle). Основная идея заключается в том, чтобы использовать ключи парами, состоящими из ключа зашифрования и ключа расшифрования, которые невозможно вычислить один из другого. В 1976 г. вышла основополагающая работа [1]. С этого времени было создано много алгоритмов, использующих концепцию открытых ключей. Алгоритм является общедоступным, нет необходимости в секретных каналах связи. Общая схема выглядит следующим образом:

1. Каждый пользователь генерирует пару ключей: один для шифрования, другой для дешифрования.

2. Каждый пользователь публикует свой ключ шифрования, размещает его в открытом для всех доступе. Второй ключ, соответствующий открытому, сохраняется в секрете.

3. Если пользователь *A* собирается послать сообщение пользователю *B*, он шифрует сообщение открытым ключом пользователя *B*.

4. Когда пользователь *B* получает сообщение, он дешифрует его с помощью своего личного (секретного) ключа. Другой получатель не сможет дешифровать сообщение, поскольку личный ключ *B* знает только *B*.

1.1. ОПИСАНИЕ АЛГОРИТМА RSA

В 1978 г. появилась работа [2], в которой Рон Райвест (Ron Rivest), Ади Шамир (Adi Shamir) и Лен Адлеман (Len Adleman) предложили алгоритм с открытым ключом. Схема Райвеста–Шамира–Адлемана (RSA) получила широкое распространение.

Опишем процесс шифрования. Исходный текст должен быть переведен в числовую форму, этот метод считается известным. В результате этого текст представляется в виде одного большого числа. Затем полученное число разбивается на части (блоки) так, чтобы каждая из них была числом в промежутке $[0, N-1]$ (о выборе N — см. ниже). Процесс шифрования одинаков для каждого блока. Поэтому мы можем считать, что блок исходного текста представлен числом x, $0 \le x \le N-1$.

Каждый абонент вырабатывает свою пару ключей. Для этого он генерирует два больших простых числа p и q, вычисляет произведение $N = p \cdot q$. Затем он вырабатывает случайное число e, взаимно простое со значением функции Эйлера от числа N, $\varphi(N) = (p-1)\cdot(q-1)$ и находит число d из условия $e \cdot d = 1 (\mathrm{mod}\ \varphi(N))$. Так как $(e, \varphi(N)) = 1$, то такое число d существует и оно единственно. Пару (N, e) он объявляет открытым ключом и помещает в открытый доступ. Пара (N, d) является секретным ключом. Для расшифрования достаточ-

но знать секретный ключ. Числа p, q, $\varphi(N)$ в дальнейшем не нужны, поэтому их можно уничтожить.

Пользователь A, отправляющий сообщение x абоненту B, выбирает из открытого каталога пару (N, e) абонента B и вычисляет шифрованное сообщение $y = x^e \pmod N$. Чтобы получить исходный текст, абонент В вычисляет $y^d \pmod N$. Так как $e \cdot d \equiv 1 (\mathrm{mod}\, \varphi(N))$, т. е. $e \cdot d = \varphi(N) \cdot k + 1$, где k – целое, то применяя теорему Эйлера, получим: следующее соотношение: $y^d \equiv (x^e)^d \equiv x^{ed} \equiv x^{\varphi(N) \cdot k + 1} \equiv (x^{\varphi(N)})^k \cdot x \equiv x (\mathrm{mod}\, N)$.

Пример 1. Пусть p = 7, q = 17. Тогда N = 7·17 = 119, $\varphi(N) = 96$. Выбираем значение e: $e < 96$, $(e, 96) = 1$. Пусть в нашем случае e = 5. Находим d: $d = 1/e \pmod{96}$. Получаем d = 77, так как 77·5 = 4·96 + 1. Открытый ключ (119,5), личный ключ (119,77). Пусть x = 19. Для зашифрования число 19 возводим в 5-ю степень по модулю 119, тогда имеем 19^5 = 2 476 099 и остаток от деления 2 476 099 на 119 равен 66. Итак, $y = 19^5 \bmod 119 = 66$, а расшифрование $x = 66^7 \bmod 119 = 19$.

О вычислениях

Как шифрование, так и расшифрование в RSA предполагают использование операции возведения целого числа в целую степень по модулю N. Если возведение в степень выполнять непосредственно с целыми числами и только потом проводить сравнение по модулю N, то промежуточные значения окажутся огромными. Здесь можно воспользоваться свойствами арифметики в классах вычетов $(a \bmod N) \cdot (b \bmod N) \bmod N = (ab) \bmod N$. Таким образом, можно рассмотреть промежуточные результаты по модулю N. Это делает вычисления практически выполнимыми.

О стойкости RSA

Безопасность алгоритма RSA основана на трудоемкости разложения на множители больших чисел. Современное состояние технических средств разложения на множители таково, что число, содержащее 193 десятичных знака, факторизовано в 2005 г. Следовательно, выбираемое N должно быть больше. Большинство общепринятых алгоритмов вычисления простых чисел p и q носят вероятностный характер.

О выборе чисел *p* и *q*

Для работы алгоритма RSA нужны простые числа. Наиболее приемлемым является генерация случайных чисел и последующая проверка их на простоту. Существуют вероятностные тесты, определяющие с заданной степенью достоверности факт простоты числа. Возникает вопрос, что произойдет, если числа окажутся составными? Можно свести вероятность такого события до приемлемого минимума, используя тесты на простоту. Кроме того, если такое событие

произойдет, это будет быстро обнаружено — шифрование и расшифрование не будут работать.

Кроме разрядности p и q, к ним предъявляются следующие дополнительные требования:

– числа не должны содержаться в списках известных больших простых чисел;

– они не должны быть близкими, так как иначе можно воспользоваться для факторизации N методом Ферма и решить уравнение $(\frac{p+q}{2})^2 - N = (\frac{p-q}{2})^2$.

– в алгоритме RSA всегда есть эквивалентные по расшифрованию показатели степеней, например d и $d' = d + [p-1, q-1]$. При этом эквивалентных решений тем больше, чем больше $(p-1, q-1)$. В лучшем случае $(p-1, q-1) = 2$, $p = 2t + 1$, $q = 2s + 1$, где s, t – нечетные числа с условием $(s, t) = 1$.

Чтобы исключить возможность применения методов факторизации накладывают следующее ограничение: числа $p-1$, $p+1$, $q-1$, $q+1$ не должны разлагаться в произведение маленьких простых множителей, должны содержать в качестве сомножителя хотя бы одно большое простое число. В 1978 г. Райвест сформулировал наиболее сильные требования. Числа $p_1 = \frac{p-1}{2}$, $p_2 = \frac{p+1}{2}$, $q_1 = \frac{q-1}{2}$, $q_2 = \frac{q+1}{2}$ должны быть простыми, причем числа $p_1 - 1$ и $q_1 - 1$ не должны разлагаться в произведение маленьких простых.

О выборе параметров *e* и *d*

Рассмотрим вопрос о выборе экспонент шифрования и расшифрования. Так как значения e и d определяют время зашифрования и расшифрования, то можно назвать ряд ситуаций, в которых желательно иметь малое значение e и d. Например, при использовании системы RSA при защите электронных платежей с применением кредитных карточек естественным является требование использования небольших значений экспоненты d у владельца карточки и большого значения экспоненты e у центрального компьютера.

Однако выбор малых параметров e или d представляется небезопасным по ряду соображений. Если малым является секретный параметр d, то можно применить метод перебора малых значений до получения искомого числа d. А если малым является параметр e, то достаточно большое число открытых сообщений, удовлетворяющих неравенству $x < \sqrt[e]{N}$, будут зашифровываться простым возведением в степень $y = x^e \pmod N$ и поэтому их можно найти путем извлечения корня степени e.

Другая аналогичная ситуация может сложиться, когда у нескольких абонентов используется одинаковая экспонента e. Тогда становится возможна атака на основе китайской теоремы об остатках (см. ниже).

Подготовка текста к шифрованию

Сначала нужно каким-либо способом представить текст сообщения в виде упорядоченного набора чисел по модулю N. Это еще не процесс шифрования, а только подготовка к нему.

Пример 2. Для простоты предположим, что текст сообщения содержит слова, записанные только заглавными буквами. Первый шаг состоит в замене каждой буквы сообщения числом. Пусть наша таблица замен имеет вид:

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С
10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27

Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
28	29	30	31	32	33	34	35	36	37	38	39	40	41

Пробел между словами будем заменять числом 99.

Например, пусть открытый текст – это девиз «ПОЗНАЙ СЕБЯ». Тогда его цифровое представление имеет вид: 25241723101999271511141.

Пусть в нашем примере $p = 149$, $q = 157$, тогда $N = 23393$. Поэтому цифровое представление открытого текста нужно разбить на блоки, меньшие, чем 23393. Одно из таких разбиений выглядит следующим образом:

$$2524 – 1723 – 10199 – 9271 – 511 – 41.$$

Конечно, выбор блоков неоднозначен, но и не совсем произволен. Например, во избежание двусмысленностей, на стадии расшифровки не следует выделять блоки, начинающиеся с нуля.

При расшифровке сообщения получаем последовательность блоков, затем их соединяем вместе и получаем число. После этого числа заменяют буквами в соответствии с таблицей, приведенной выше.

Обратим внимание на то, что в этом примере каждую букву кодируем двузначным числом. Это сделано для предотвращения неоднозначности. Если бы мы пронумеровали буквы не по порядку, начиная с 1, т. е. А соответствует 1, Б соответствует 2 и т. д., то было бы непонятно, что обозначает блок 12: пару букв АБ или букву Л, двенадцатую букву алфавита. Конечно, для кодирования можно использовать любые однозначные соответствия между буквами и числами, например ASCII-кодировку, что чаще всего это и делается.

Продолжим пример: выбираем $p = 149$, $q = 157$, вычисляем $\varphi(N) = 23\,088$. Теперь нужно выбрать число e, взаимно простое с $\varphi(N)$. Наименьшее простое, не делящее $\varphi(N)$, равно 5. Положим $e = 5$. Зашифруем первый блок сообщения: вычисляем $2524^5 \bmod 23393 = 22752$; далее $1723^5 \bmod 23393 = 6198$.

$10199^5 \bmod 23393 = 14204$,

$9271^5 \bmod 23393 = 23191$,

$511^5 \bmod 23393 = 10723$,

$41^5 \bmod 23393 = 14065$.

Теперь шифрованный текст имеет вид

$$22752619814204231911072314065$$