УДК 681.324

К. А. АБДИКАЛИКОВ

МОДИФИЦИРОВАННАЯ КРИПТОГРАФИЧЕСКАЯ СИСТЕМА НА ОСНОВЕ АЛГОРИТМА RSA-MOD

Рассматриваются эффективные алгоритмы компьютерные технологии решения задачи защиты информации на базе алгоритма RSA.

При решении практических задач нередко приходится выполнять операции над целыми числами большой разрядности, т. е. числами, для размещения которых требуется n ячеек памяти компьютера. Такие числа будем называть еще многоразрядными или n -словными. Если n большое, то выполнение некоторых операций требует существенных затрат машинного времени и возникает необходимость в оптимизации по быстродействию соответствующих алгоритмов.

При реализации криптографических систем защиты информации, основанных на методах RSA [1], основная вычислительная нагрузка приходится на выполнение арифметических операций над многоразрядными числами.

Опишем компьютерные технологии модифицированного алгоритма RSA-MOD.

1 шаг. Пусть P и Q – два различных больших случайно выбранных простых числа. Для определения двух больших случайных простых чисел P и Q произвольно выбирается нечетное целое число r подходящего размера (например, 100–200 разрядов) и проверяется на простоту. В тестах для проверки чисел на простоту используется алгоритм Рабина-Миллера [2].

2 шаг. После выбора P и Q вычисляем

$$N = P \cdot Q. \qquad (1)$$

Чтобы найти произведение (1), используется алгоритм типа Шенхаге–Штрассена [3].

В алгоритме перемножаемые числа P и Q представляются в виде полиномов:

$$P = \sum_{i=0}^{k-1} P_i \cdot 2^i, \quad Q = \sum_{j=0}^{k-1} Q_j \cdot 2^j,$$

где числа P_i – это блоки, составленные из l разрядов числа P; Q_l – блоки из l разрядов числа Q. Алгоритм вычисления произведения $t_{k-1}, t_{k-2}, \text{K}, t_0$ двух целых чисел, основанный на процедуре быстрого преобразования Фурье [4], выглядит следующим образом:

Шаг 2.1. С помощью процедуры БПФ вычислить $P(w^i)$ и $Q(w^i)$ для $i = 0, 1, ..., k–1$, где w – примитивный k-й корень из единицы.

Шаг 2.2. Перемножить почленно

$$(P(1)Q(1), P(w)Q(w), \text{K}, P(w^{i-1})Q(w^{i-1})).$$

Шаг 2.3. Интерполировать $t = \sum_{i=0}^{k-1} t_i 2^i$, вычисляя

$$k^{-1} \cdot \sum_{i=0}^{k-1} P(w^i)Q(w^i) \cdot 2^i,$$

для $\{1, w^{-1}, \text{K}, w^{-(k-1)}\}$ с помощью процедуры ОБПФ.

Шаг 2.4. Возвратить коэффициенты $t_{i-1}, t_{i-2}, \text{K}, t_0$.

Шаг 3. Открытый ключ E выбирают случайным образом так, чтобы выполнялись условия:

$$1 \le E \le \varphi(N), \; НОД(E, \varphi(N)) = 1, \qquad (2)$$

$$\varphi(N) = (P-1)(Q-1), \qquad (3)$$

где $\varphi(N)$ – функция Эйлера.

Шаг 4. Далее, используя расширенный алгоритм Евклида, вычисляется секретный ключ D такой, что

$$D < N, \quad E \cdot D \equiv 1 (\text{mod}\, \varphi(N)). \qquad (4)$$

При заданных неотрицательных целых числах E и D алгоритм определяет вектор (u_1, u_2, u_3) такой, что

$$E \cdot u_1 + D \cdot u_2 = u_3 = НОД(E, D).$$

В процессе вычисления используются вспомогательные векторы (v_1, v_2, v_3), (t_1, t_2, t_3).

Действия с векторами проводятся таким образом, что в течение всего процесса вычисления выполняются соотношения

$$E \cdot t_1 + D \cdot t_2 = t_3,$$

$$a \cdot u_1 + D \pm \cdot u_2 = u_3, \quad a \cdot v_1 + D \cdot v_2 = v_3.$$

Вычисление D (4) проводится следующим образом:

Шаг 4.1. Установить $(u_1, u_2, u_3) := (0, 1, N)$,

$$(v_1, v_2, v_3) := (1, 0, E).$$

Шаг 4.2. $u_3 = 1$? Если $u_3 = 1$, то алгоритм завершает свою работу.

Шаг 4.3. Установить $q := \left[u_3 / v_3 \right]$,

$$(t_1,\ t_2,\ t_3) := (u_1,\ u_2,\ u_3) - (v_1,\ v_2,\ v_3) \cdot q,$$

$$(u_1,\ u_2,\ u_3) := (v_1,\ v_2,\ v_3),$$

$$(v_1,\ v_2,\ v_3) := (t_1,\ t_2,\ t_3).$$

Шаг 4.4. Возвратиться к шагу 4.2.

Пара чисел (E, N) является открытым ключом.

Шаг 5. Пользователь **B** пересылает пользователю **A** пару чисел (E, N) по незащищенному каналу. Если пользователь **A** желает передать пользователю **B** сообщение M, выполняет следующие шаги:

Шаг 5.1. Пользователь **A** разбивает исходный текст M на секции, каждый из которых может быть представлен виде $M_i = \left[\log_2(n)\right]$.

Шаг 5.2. Пользователь **A** шифрует текст, представленный в виде последовательности чисел M_i по формуле

$$C_i = M_i^E \pmod N. \qquad (5)$$

Шаг 5.3. Отправляет криптограмму $C_i = C_1, C_2, C_3, \ldots, C_i$ пользователю **B**.

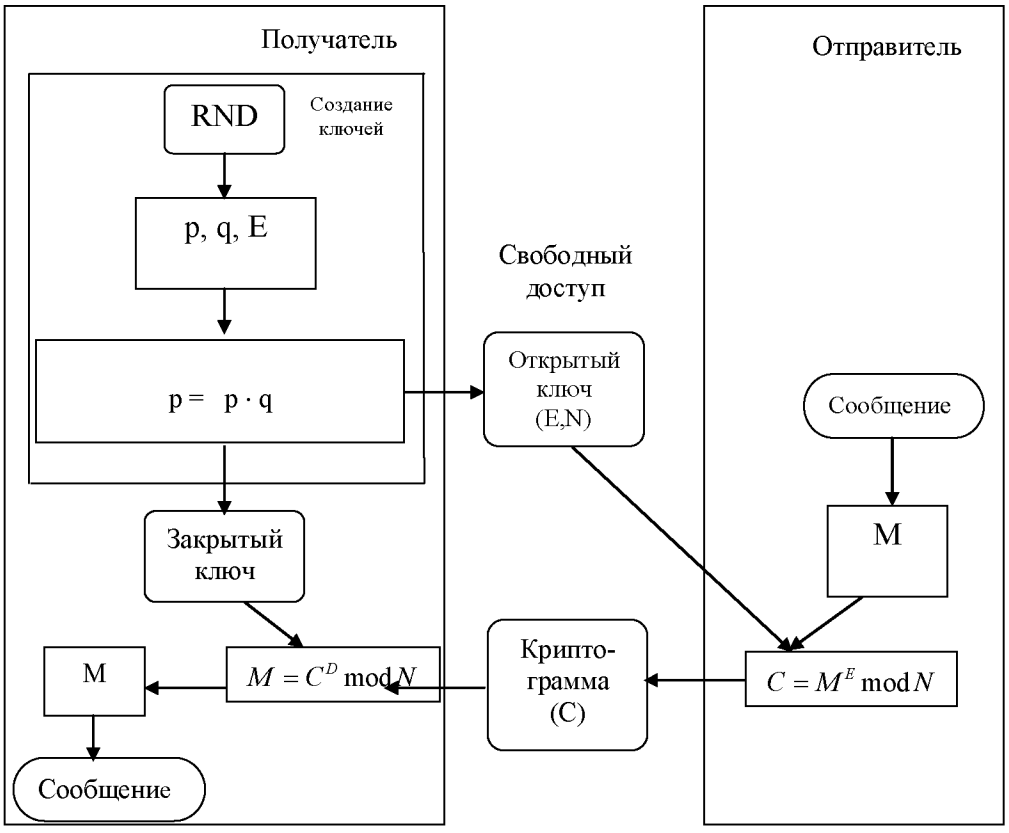
Шаг 6. Пользователь **B** расшифровывает принятую криптограмму

$$C_i = C_1, C_2, C_3, \ldots, C_i,$$

используя секретный ключ D, по формуле

$$M_i = C_i^D \pmod N. \qquad (6)$$

В результате будет получена последовательность чисел M_i, которая представляет собой исходное сообщение M.



Модифицированная схема RSA-MOD

Операцией, необходимой при зашифровании и расшифровании, является модульное возведение в степень. Это можно сделать более быстрее, чем в известных криптографических системах [1]. В качестве алгоритма быстрого вычисления (5) и (6) используются алгоритм Монтгомери и бинарный метод [5], а также можно использовать китайскую теорему об остатках [6, 7].

Бинарный метод с использованием метода Монтгомери будет определяться следующим образом:

Входные данные: N, E, N, R, n.

Выходные данные C.

Шаг 1. Вычислить $N' = N^{-1} \bmod R$ с помощью расширенного алгоритма Евклида.

Шаг 2. $\overline{M} = M \cdot R \bmod N$.

Шаг 3. Если $E = 1$, то $\overline{X} = \overline{M}$, иначе $\overline{X} = 1 \cdot R \bmod N$.

Шаг 4. Для $i = k$–1 до 0 выполнить

Шаг 5. $\overline{X} := Monpro(X, \overline{X})$.

Шаг 5.1. $t = \overline{a} \cdot \overline{b}$;

Шаг 5.2. $M = tN' \bmod R$;

Шаг 5.3. $u = (t + M \cdot N)/R$.

Шаг 5.4. Если $u \geq N$, то вычислить $u - N$, иначе шаг 5.3.

Шаг 6. Если E: = 1, то $\overline{X} := Monpro(\overline{M}, \overline{X})$.

Шаг 7. $C := Monpro(\overline{X}, 1)$.

Модифицированная схема RSA-MOD приведена на рис.

Таким образом, начинаем с вычисления обычного вычета от M и получаем N – вычет $\overline{M}$, используя операцию подобную операции деления, чего можно достичь, например, выполняя серию сдвигов и вычитаний. Кроме того, шаги 2 и 3 требуют деления. Однако как только предвычисления на этом завершаются, внутренний цикл бинарного метода вычисления степени использует операцию произведения Монтгомери, которая выполняется только с помощью умножения по модулю 2^k и деления на 2^k. Когда завершает свою работу бинарный метод, получаем N – вычет величины $\overline{X}$ от значения $C = M^E \bmod N$.

Таким образом, рассмотрена новая эффективная по быстродействию криптосистема RSA-MOD.

ЛИТЕРАТУРА

1. *Rivest R.L., Shamir A, and Adleman.* A method for obtaining digital signatures and public-key cryptosystems // Comm. ACM. 1978. N 21. P. 120-126.

2. *Абдикаликов К.А.* Анализ вычислительной сложности алгоритмов тестирования чисел на простоту // Поиск. 2002. № 4(2). С. 145-150.

3. *Шенхаге А., Штрассен В.* Быстрое умножение больших чисел // Кибернет. сб. 1973. Вып. 10. С. 87-98.

4. *Задирака В.К., Абдикаликов К.А.* Быстрые ортогональные преобразования: теория и приложения. Алматы: Fылым, 2003. 220 с.

5. *Montgomery P.L.* Modular Multiplication Without Trial Division // Math. Comp. 1985. V. 44, N 170. P. 519-521.

6. *Акушский И.Я., Амербаев В.М., Пак И.Т.* Основы машинной арифметики комплексных чисел. Алма-Ата: Наука, 1970. 248 с.

7. *Абдикаликов К.А.* Быстрое шифрование с использованием китайской теоремы об остатках // Вестник МОН РК, НАН РК. 2004. № 3. С. 144-1149.

Резюме

RSA алгоритмі негізінде ақпараттық қауіпсіздік есептері шешімдерінің тиімді алгоритмдерінің компьютерлік технологиясы қарастырылады.

Summary

Effective algorithms of computer technologies of the decision of a problem of information safety are considered on the basis of algorithm RSA

Актюбинский госуниверситет им. К. Жубанова МОН РК, г. Актобе

Поступила 20.06.06 г.