УДК 581.327

МЕТОДЫ КРИПТОАНАЛИЗА БЛОЧНЫХ ШИФРОВ И КРИПТОСИСТЕМЫ RSA

А.М. Ткачев
(представил д.т.н., проф. Ю.В. Стасев)

В статье описаны различные виды атак на одну из наиболее популярных ныне схем открытого шифрования и цифровой подписи - RSA. Рассматриваемые автором методы предполагают наличие определенных математических слабостей схемы, не учитываемых при реализации атаки на систему. Также приводятся меры противодействия этим атакам. Их необходимо принимать во внимание при реализации схемы RSA или протоколов, основанных на ней.

RSA - асимметричная криптосистема в настоящее время является широко распространенной в мире и называется так по первым буквам в именах создателей R. Rivest, A. Shamir, L. Adleman.

1. Описание криптосистемы [3, 5]. Для применения криптосистемы первоначально необходимо выполнить следующую последовательность действий:

1) выбираются **p**, **q** – большие простые числа; вычисляется произведение

$$\mathbf{n = p \cdot q};$$

2) выбирается число **e** – такое, что

$$\mathbf{(e, \varphi(n)) = 1},$$

(**e** и **φ** (**n**) – взаимно просты), где **φ** (**n**) - функция Эйлера от **n**;

3) из уравнения

$$\mathbf{ed = 1 \pmod{\varphi(n)}}$$

находится число **d**.

Полученные числа **e**, **n** – открытый ключ пользователя, а **d** – секретный ключ.

Процедура зашифрования

$$\mathbf{C = E_{(e,n)}(M) = M^e \pmod{n}},$$

где **C** – получаемый шифротекст; **M** – открытый текст, удовлетворяющий следующему условию

$$\mathbf{M^{\varphi(n)} = 1 \pmod{n}}.$$

Процедура расшифрования:

$$\mathbf{M = D_{(d,n)}(C) = C^d (mod\ n)}.$$

Генерация цифровой подписи

$$\mathbf{Q = M^d (mod\ n)}.$$

Проверка цифровой подписи

$$\mathbf{Q^e (mod\ n) = M}.$$

Оценим вероятность того, что сообщение будет не взаимнопросто с **n**, т.е. $\mathbf{(m, n) \neq 1}$. Число всех чисел – **n**; число чисел, взаимно простых с **n** – $\boldsymbol{\varphi}$(**n**). Значит вероятность попадания **m** в совокупность чисел, не взаимопростых с **n** [8]:

$$\mathbf{P\{(m, n) \neq 1\} = (n - \varphi(n))/n = (pq - (p-1)(q-1)/pq < 1/p + 1/q}.$$

Рассмотрим стойкость криптосистемы и возможные атаки.

Стойкость RSA основывается на проблеме факторизации больших простых чисел. Действительно, если злоумышленнику удастся разложить **n** на делители **p** и **q**, то для него не составит труда вычислить **n**, а затем определить секретный ключ пользователя. Однако нахождение секретного ключа RSA не эквивалентно проблеме факторизации. Это означает, что

$$\mathbf{T(RSA) \leq T(факторизации)},$$

где **T**(RSA) – трудоемкость определения секретного ключа RSA, а **T**(факторизации) – трудоемкость факторизации числа **n** [10].

Таким образом могут быть найдены эффективные алгоритмы определения секретного ключа RSA, причем в то же время проблема факторизации не будет разрешена.

Рассмотрим основные методы криптоанализа блочных шифров и криптосистемы RSA.

2. Метод встречи [3]. Данный метод применяется для атаки на блочные шифры. Обладает значительно меньшей трудоемкостью по сравнению с методом полного перебора.

Даны открытый и шифрованный тексты. Криптосистема состоит из **h** циклов шифрования. Цикловые ключи независимы и не имеют общих битов. Ключ **K** системы представляет собой сочетание из **h**-цикловых ключей $\mathbf{k_1}, \mathbf{k_2}$, ..., $\mathbf{k_n}$.

Необходимо при известных открытом и шифрованном текстах найти ключ **K**.

Обозначим преобразование алгоритма как

$$\mathbf{E_k(a) = b},$$

где **a** – открытый текст, а **b** – шифротекст.

Его можно представить как композицию

$$\mathbf{E_{k1} E_{k2} ... E_{kh}(a) = b},$$

где $\mathbf{E_{ki}}$ – цикловое преобразование на ключе $\mathbf{k_i}$. Каждый ключ $\mathbf{k_i}$ пред-

ставляет собой двоичный вектор длины **n**, а общий ключ системы – вектор длины $\mathbf{n \times h}$.

Будем перебирать все значения $\mathbf{k' = (k_1, k_2, ..., k_r)}$, т.е. первые **r** цикловых ключей. На каждом таком ключе **k'** зашифровываем открытый текст

$$\mathbf{a - E_{k'}(a) = E_{k1} E_{k2} .. E_{kr}(a) = S},$$

(т.е. проходим **r** циклов шифрования вместо **h**). Будем считать **S** неким адресом памяти и по этому адресу запишем значение **k'**. Необходимо перебрать все значения **k'**.

Перебираем все возможные $\mathbf{k' = (k_{r+1}, k_{r+2}, \ldots, k_h)}$. На получаемых ключах расшифровываем шифротекст [5]

$$\mathbf{b - E^{-1}_{k"}(b) = E^{-1}_{kh}, ..., E^{-1}_{kr+1}(b) = S'}.$$

Если по адресу **S'** не пусто, то достаем оттуда ключ **k'** и получаем кандидат в ключи

$$\mathbf{(k', k'') = k}.$$

Однако нужно заметить, что первый же полученный кандидат **k** не обязательно является истинным ключом. Для данного открытого текста **a** и шифротекста **b** выполняется $\mathbf{E_k(a) = b}$, но на других значениях открытого текста **a'** шифротекста **b'**, полученного из **a'** на истинном ключе, равенство может нарушаться. Все зависит от конкретных характеристик криптосистемы. Но иногда бывает достаточно получить такой "псевдоэквивалентный" ключ. В противном же случае после завершения процедур будет получено некое множество ключей **{k', k", ...}**, среди которых находится истинный ключ.

Если рассматривать конкретное применение, то шифротекст и открытый текст могут быть большого объема (например, графические файлы) и представлять собой достаточно большое число блоков для блочного шифра. В данном случае для ускорения процесса можно зашифровывать и расшифровывать не весь текст, а только его первый блок (что намного быстрее) и затем, получив множество кандидатов, искать в нем истинный ключ, проверяя его на остальных блоках.

3. Метод безключевого чтения RSA [4].

Противнику известны открытый ключ (**e**, **n**) и шифротекст **C**. Необходимо найти исходный текст **M**.

Противник подбирает число **j**, для которого выполняется следующее соотношение:

$$\mathbf{C^{ij} (mod\ n) = C},$$

т.е. противник просто проводит **j** раз зашифрование на открытом ключе перехваченного шифротекста. Это выглядит следующим образом:

$$\mathbf{(C^e)^e)^e \ldots)^e (mod\ n) = C^{ej} (mod\ n)}.$$

Найдя такое **j**, противник вычисляет $\mathbf{C^{ej-1} (mod\ n)}$ (т.е. **j**–1 раз по-

вторяет операцию зашифрования) – это значение и есть открытый текст **M.** Это следует из того, что некоторое число $\mathbf{C^{ej-1}(mod\ n)}$ в степени **e** дает шифротекст **C**.

Пример: **p = 983**; **q = 563**; **e = 49**; **M = 123456**. Тогда **C = M49(mod n)**; **C497(mod n) = 85978**; **C498(mod n) = 123456**; **C499(mod n) = 1603**.

4. Атака на подпись RSA в схеме с нотариусом [9].

Имеется электронный нотариус, подписывающий проходящие через него документы. **N** – некоторый открытый текст, который нотариус не желает подписывать. Противнику известен открытый ключ (**e**, **n**) нотариуса. Необходимо подписать этот текст **N**.

Противник вырабатывает некое случайное число **x**, которое взаимно просто с **N** и вычисляет

$$\mathbf{y = xe(mod\ n)},$$

затем получает значение

$$\mathbf{M = yN}$$

и передает его на подпись нотариусу. Тот подписывает (ведь это уже не текст **N**)

$$\mathbf{Md\ (mod\ n) = S}.$$

Следовательно, получаем, что

$$\mathbf{S = Md\ (mod\ n) = ydNd = (xe)\,dNd = xNd},$$

а значит

$$\mathbf{Nd = Sx - 1\ (mod\ n)}.$$

Это означает, что надо разделить полученное **S** на **x**.

Для обеспечения защиты при подписи необходимо добавлять некоторое случайное число в сообщение (например, время). Таким образом получится искажение числа **M** при подписи, т.е

$$\mathbf{M_{(после\ добавления)} \neq yN}.$$

5. Атака на подпись RSA по выбранному шифротексту [4]. Пусть имеется шифротекст **C**. Противнику известен открытый ключ (**e**, **n**) отправителя сообщения. Необходимо найти исходный текст **M**.

Противник вырабатывает некое **r**, что **r** < **n**, (**r**, **n**) = 1 и вычисляет

$$\mathbf{x = re\ (mod\ n)}.$$

Затем он вычисляет

$$\mathbf{t = r - 1\ (mod\ n)};$$
$$\mathbf{y = xC(mod\ n)}$$

и посылает **y** на подпись отправителю.

Отправитель, ничего не подозревая, подписывает текст **y**:

$$w = yd \pmod n$$

и отправляет **w** обратно.

Противник вычисляет

$$tw \pmod n = r - lyd \pmod n = (\text{т.к. } r = xd \bmod n) =$$

$$= x - dxdCd \pmod n = Cd = M.$$

Противник не может сразу послать **C** на подпись, т.к. отправитель просматривает полученные в результате подписи сообщения и может заметить провокацию.

Атака носит несколько гипотетический характер, но тем не менее позволяет сделать несколько важных выводов: а) подписывать и шифровать надо разными ключами, либо б) добавлять случайный вектор при подписи или использовать хэш-функцию.

ЛИТЕРАТУРА

1. *Bruce Schneier Applied Cryptography: protocols, algorithms and source codes in C. John Wiley & Sons, Inc. 1994. – 564 p.*
2. *Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone Handbook of Applied Cryptography. CRC Press, 1996. – P. 264 - 277.*
3. *Ростовцев А.Г., Матвеев В.А. Защита информации в компьютерных системах. Выпуск 2: Элементы криптологии. – С-Пб., изд-во СПбГТУ, 1993. – 376 с.*
4. *Молдовян Н.А. Проблематика и методы криптографии. – С-Пб., изд-во СПбГУ, 1998. – 290 с.*
5. *Молдовян Н.А. Скоростные блочные шифры. – С-Пб., изд-во СПбГУ, 1998. – 412с.*
6. *David Kahn The Codebreakers: The Story of Secret Writing. New York: MacMillan, 1996. – 320 p.*
7. *Edited by J.van Leeuwen Handbook of Theoretical Computer Science, chapter 13: Cryptography (by Ronald L.Rivest) Elsevier Science Publishers B.V., Holland, 1990. – P. 211-223.*
8. *Thomas H.Cormen, Charles E.Leiserson, Ronald L.Rivest Introduction to Algorithms, paragraphs 33.7-33.9 MIT Press, 1990. – P. 501 - 513.*
9. *Ross J. Anderson Why Cryptosystems Fail Communications of the ACM, Vol.37, No.11, Novemvber 1994. – P.32-40.*
10. *C.Charnes, L.O'Connor, J.Pieprzyk, R.Safavi-Naini, Y.Zheng Further Comments on the Soviet Encryption Algorithm, 1994. – 110 p.*
11. *Odlyzko A. The future of integer factorization CryptoBytes (The technical newsletter of RSA Laboratories), 1 (no. 2), 1995. – P. 5 - 12.*

Поступила 13.05.2002

ТКАЧЕВ Андрей Михайлович, *преподаватель ХВУ. В 1996 году окончил ХВУ. Область научных интересов - системы защиты информации.*