КРИПТОСИСТЕМА RSA

Чичикин Г.Я.[1], Семёнов Д.А.[2]

[1]Чичикин Гордей Ярославович – студент;
[2]Семёнов Дмитрий Андреевич – студент,
кафедра защиты информации,
Институт комплексной безопасности и специального приборостроения,
Российский технологический университет,
г. Москва

Аннотация: *в данной статье рассматривается ассиметричное шифрование RSA, которая основывается на сложности задачи факторинга. Этот алгоритм по сей день используется в различных криптографических приложениях. Эта криптосистема первая стала пригодна для надежного шифрования и цифровой подписи одновременно. Особенность этого алгоритма в том, что открытые ключи можно передавать по незащищенным каналам связи.*

Ключевые слова: *RSA, шифрование, факторизация, цифровая подпись.*

УДК 004.031.2

RSA (Rivest-Shamir-Adleman) является одной из первых криптосистем с открытым ключом, используемый для безопасной передачи данных. В криптосистеме ключ шифрования является открытым (public) и отличается от ключа дешифрования тем, что хранится в секрете (private). Ассиметрия в RSA основана на практической трудности факториала произведения двух больших простых чисел, "задачи факторинга". Аббревиатура состоит из начальных букв фамилий Рона Ривеста (Rivest), Ади Шамира(Shamir) и Леонарда Адлемана (Adelman), которые впервые опубликовали алгоритм в 1978 году

Пользователь RSA создает и публикует открытый ключ на основе двух больших простых чисел, а также вспомогательное значение. Простые числа должны быть засекречены. Любой может использовать открытый ключ для шифрования сообщения.

Из-за этого он реже используется для прямого шифрования пользовательских данных. Чаще всего RSA передает зашифрованные общие ключи для симметричной криптографии ключей, которые, в свою очередь, могут выполнять массовые операции шифрования-дешифрования с гораздо большей скоростью.

Идея асимметричной криптосистемы с открытым и закрытым ключами принадлежит Уитфилду Диффи и Мартину Хеллману, которые опубликовали эту концепцию в 1976 году. **[1]**

Основным принципом RSA является наблюдение, что практически найти три очень больших положительных целых числа *e, d* и *n*, таких, что с модулярной экспоненцией для всех целых чисел *m (с* $0 \le m < n$):

$$(\mathrm{m}^{\mathrm{e}})^{\mathrm{d}} = \mathrm{m}(\mathrm{mod}\,\mathrm{n})$$

и что, даже зная e и n или даже m, может быть чрезвычайно трудно найти d.Кроме того, для некоторых операций удобно, что порядок двух экспонент может быть изменен и что это отношение также подразумевает:

$$\left(\mathrm{m}^{\mathrm{d}}\right)^{\mathrm{e}} = \mathrm{m}(\mathrm{mod}\,\mathrm{n})$$

Ключи для алгоритма RSA генерируются следующим образом:

1. Выбрать два различных простых числа *p* и *q*.

- В целях безопасности целые числа *p* и *q* следует выбирать случайным образом, и они должны быть одинаковыми по величине, но отличаться по длине на несколько цифр, чтобы сделать факторинг сложнее. Простые целые числа можно эффективно найти с помощью теста на примитивность.

2. Вычислить $n = pq$.

- n используется в качестве модуля как для открытого, так и для закрытого ключей. Его длина, обычно выражаемая в битах, является длиной ключа.

3. Вычислить $\lambda(n) = НОК(\varphi(p), \varphi(q)) = НОК(p-1, q-1)$ = где λ - функция Кармайкла. Это значение хранится в секрете.

4. Выберите целое число e, такое, что $1 < e < \lambda(n)$,$НОД(e, \lambda(n)) = 1$; т.е. e и $\lambda(n)$ являются копримами.

5. Определить d как $d \equiv e - 1 (\mathrm{mod}\ \lambda(n))$; то есть d-модулярный мультипликативный обратный от e по модулю $\lambda(n)$.

Это означает: решить для D уравнение $d \cdot e \equiv 1 (\mathrm{mod}\ \lambda(n))$.

Наличие e короткой длины бита и небольшого веса Хэмминга приводит к более эффективному шифрованию – чаще всего $e = 216 + 1 = 65{,}537$. Однако в некоторых настройках было показано, что гораздо меньшие значения e (например, 3) менее безопасны. e освобождается как показатель открытого ключа.D хранится как показатель частного ключа. Открытый ключ состоит из модуля n и открытого (или шифрования) показателя e. Закрытый ключ состоит из частного (или расшифровочного) показателя D, который должен храниться в секрете. p, q и $\lambda(n)$ также должны храниться в секрете, поскольку их можно использовать для вычисления d.

В оригинальной работе RSA для вычисления частного показателя d используется функция Эйлера e вместо λ(n). Поскольку $\varphi(n)$ всегда делится на $\lambda(n)$, алгоритм также работает. Таким образом, любое d, удовлетворяющее $d \cdot e \equiv 1 (\mathrm{mod}\ \varphi(n))$, также удовлетворяет $d \cdot e \equiv (\mathrm{mod} \lambda(n))$. Однако вычисление d по модулю $\varphi(n)$ иногда дает результат, который больше, чем необходимо (например, $d > \lambda(n)$). Любые "негабаритные" частные показатели, не удовлетворяющие этому критерию, всегда могут быть уменьшены по модулю $\lambda(n)$ для получения меньшего эквивалентного показателя.

Поскольку любые общие факторы $(p-1)$ и $(q-1)$ присутствуют в факторизации $n - 1 = pq - 1 = (p-1)(q-1) + (p-1) + (q-1)$, рекомендуется, чтобы $(p-1)$ и $(q-1)$ имели только очень малые общие факторы, если они есть, кроме необходимых двух.

Предположим, Боб хочет послать информацию Алисе. Если они решат использовать RSA, Боб должен знать открытый ключ Алисы для шифрования сообщения, а Алиса должна использовать свой закрытый ключ для расшифровки сообщения. Чтобы Боб мог отправлять свои зашифрованные сообщения, Алиса передает Бобу свой открытый ключ (n, e) надежным, но не обязательно секретным путем. Закрытый ключ Алисы (d) никогда не распространяется.

После того, как Боб получит открытый ключ Алисы, он может отправить сообщение M Алисе. Для этого он сначала превращает M в целое число m, такое, что $0 \leq m < n$, используя согласованный обратимый протокол, известный как схема заполнения. Затем он вычисляет шифротекст c, используя открытый ключ Алисы e, соответствующий:$c \equiv m^e \mathrm{mod}(n)$

Это можно сделать достаточно быстро, даже для 500-битных чисел, используя модульное возведение в степень.

Алиса может восстановить m из c, используя ее показатель частного ключа d путем вычисления: $c^d \equiv (m^e)^d \equiv m (\mathrm{mod}\ n)$

Учитывая m, она может восстановить исходное сообщение M, изменив схему заполнения

Вот пример шифрования и дешифрования RSA. Параметры, используемые здесь, искусственно малы, но можно также использовать OpenSSL для генерации и изучения реальной пары клавиш.

1. Выберите два различных простых числа, например $p = 61$ и $q = 53$
2. Вычислить $n = 61 * 53 = 3233$ давать
3. Вычислить функцию Кармайкла, $\lambda(n) = НОК(П - 1, м - 1)$ дает
4. Выбрать любое число $1 < e < 780$, которое соответствует 780. Выбор e в качестве простого числа оставляет нам только проверить, что e не является делителем 780.

Пусть e=17,

5. Вычислить d, $e \pmod{\lambda(n)}$, дающий, $d = 413\ d * e = 1 \bmod \lambda(n)$, $413 * 17 = 1 \bmod 780$Открытый ключ $(n = 3233, e = 17)$. Для дополненного сообщения открытого текста m функция шифрования $c(m) = m^{17} \bmod 32\,33$

Закрытый ключ (n = 3233,d = 413)Для зашифрованного шифротекста c функция расшифровки $m(c) = c^{413} \bmod 3233$ Например, чтобы зашифровать m = 65, мы вычисляем : $c = 6517 \bmod 3233 = 2790$. Чтобы расшифровать c = 2790, мы вычисляем :$m = 2790^{413} \bmod 3233 = 65$

Оба эти расчета могут быть эффективно вычислены с помощью квадрата и умножены по алгоритму модульного возведения в степень. В реальных ситуациях выбранные простые числа были бы намного больше; в нашем примере было бы тривиально разложить N, 3233 (полученные из открытого ключа) обратно на простые числа p и q. e, также из открытого ключа, затем инвертируется, чтобы получить d, таким образом, приобретая закрытый ключ.

Список литературы

1. [Электронный ресурс]. Режим доступа: http://people.csail.mit.edu/rivest/Rsapaper.pdf (дата обращения 18.04.2019).
2. [Электронный ресурс]. Режим доступа: https://people.csail.mit.edu/rivest/pubs/ARS03.rivest-slides.pdf (дата обращения 30.04.2019).