АНАЛИЗ АЛГОРИТМА ШИФРОВАНИЯ NTRU СОГЛАСНО СТАНДАРТА ANSI X9.98

Беликова Е.С., Заросилова М.Г.
Харьковский национальный университет радиоэлектроники
61166, Харьков, пр. Ленина, 14, каф. Безопасности Информационных Технологий
тел. (057) 702-14-25 E-mail: elenochka.flame@gmail.com

This work is devoted to analysis of encryption algorithm NTRU. Here it is given a describing of key generetion, encryption and decription procedures; advantages of NTRU, such as perfomans, durability to quantum computer attack.

Существует множество ассиметричных систем. Одна из наиболее распространенных асимметричных криптосистем — RSA. RSA плохо подходит для использования во встраиваемых устройствах, потому что требует оперирования большими числами, что не всегда можно реализовать при ограниченных ресурсах. Кроме того, RSA работает довольно медленно, особенно это касается генерации ключей, и в RSA используются более длинные ключи по сравнению с другими криптосистемами с открытым ключом. Но RSA хорошо подходит для сравнения криптографической стойкости других криптосистем. Для использования криптографии в мобильных устройствах, память и вычислительная мощность которых сильно ограничены, необходимы более эффективные алгоритмы. Самая распространенная из таких криптосистем — эллиптическая криптосистема, которая требует меньше ресурсов, чем RSA. Хотя эллиптическая криптосистема не настолько хорошо исследована, как RSA, она считается надежной и используется в нескольких стандартах. Криптосистема, созданная позднее, такая как: NTRU, работают быстрее, чем эллиптическая криптосистема, но ее надежность недостаточно исследована.

NTRU был разработан в середине 1990-х годов и впервые был представлен на конференции CRYPTO'96. В этом алгоритме все операции производятся в кольце усечённых многочленов. Криптографическая стойкость алгоритма основана на сложности задачи нахождения короткого вектора в заданной решётке.

Схема шифрования SVES состоит из пяти операций генерации ключа, подтверждения правильности ключевой пары, подтверждения правильности открытого ключа, зашифрования и расшифрования.

Генерация ключа. Для заданного набора общесистемных параметров ассиметричная пара состоит из личного ключа f и открытого ключа h, которые являются полиномами степени N-1.

Ключевые пары тесно связаны с доменными параметрами и должны использоваться только вместе с конкретными общесистемными параметрами, при участии которых были сгенерированы.

Личный ключ f и вспомогательный полином g в общем случае генерируются случайным (псевдослучайным) образом лишь с тем условием, что для f должны существовать обратные полиномы как по модулю p, так и по модулю q. Для стандарта X9.98 он должен иметь вид f = 1 + p*F, где F генерируется с помощью генератора случайных чисел, p — меньший модуль, целое число (в данном стандарте p = 3).

Ключи генерируются или с использованием генератора случайных бит RBG в связке с индексной функцией генерации IGF, или с помощью генератора случайных чисел (диапазон значений от 0 до N-1). Чтобы ключи соответствовали уровню безопасности в k бит, генератор случайных чисел/бит должен быть проинициализирован хотя бы 64+k битами энтропии (т. е. выполнить 64+k холостых циклов генерации).

Шифрование. Для осуществления процедуры зашифрования необходимо сгенерировать полином r. В общем случае полином r выбирается случайным (псевдослучайным) образом (со степенью, не большей N-1). В стандарте X9.98 он детерминировано формируется из сообщения m и случайного значения *b* с помощью псевдослучайного генератора.

Процедура зашифрования определяется следующей формулой:

$$e = r*h + m \pmod q,$$

где r – случайный полином,
h- открытый ключ,
m – сообщение,
q – больший модуль (в стандарте X9.98 q=2048)
e — криптограмма.

Расшифрование. Расшифрование выполняется следующим образом. Сначала вычисляется многочлен a:

$$a = f*e \bmod q,$$

где f – секретный ключ,
e — криптограмма,
q – больший модуль (в стандарте X9.98 q=2048).

В стандарте X9.98 коэффициенты многочлена a лежат в пределах [A, A + q -1], A — общесистемный параметр, зависит от остальных ОСП, обычно большое отрицательное число.

Далее вычисляется

$$m^* = f_p^{-1}*a \bmod p,$$

где f_p^{-1} – обратный многочлен по модулю p.

Расшифрованное сообщение m*, в общем случае, может не совпадать с исходным. Это происходит из-за того, что операции производятся сначала по модулю q, а потом по модулю p. Поэтому необходимо, чтобы многочлен a имел коэффициенты из такого интервала, чтобы их не пришлось приводить по модулю q. Это позволит восстановить исходное сообщение.

NTRU имеет два основных преимущества перед другими ассиметричными алгоритмами:

1. NTRU работает быстрее, чем используемые в настоящее время криптосистемы с открытым ключом. Скорость работы NTRU гораздо выше, чем RSA и EC: он в 1300 раз быстрее 2048-битного RSA и в 117 раз быстрее ECC NIST-224 (если сравнивать количество операций в секунду), или в 1113 раз быстрее, чем 2048-битный RSA (если сравнивать пропускную способность).

2. Если задача факторизации целых чисел, задача дискретного логарифма в конечных полях и задача дискретного алгоритма на эллиптических кривых были решены (например, реализацией квантового компьютера достаточного размера), задача нахождения короткого вектора в решетке является криптостойкой к алгоритмам, выполняемым на квантовых компьютерах.