УДК 519.6

М. В. Белоусов, А. В. Александров

ОСОБЕННОСТИ РЕАЛИЗАЦИИ SMT-ПРОТОКОЛА НА БАЗЕ ЯЗЫКА PYTHON 3

Описана реализация SMT-протокола на основе схемы разделения секретного сообщения Шамира, исследованы характеристики протокола, такие как алгоритмическая сложность, скорость работы и надежность.

Ключевые слова: *SMT-протокол, схема разделения секрета, модель Долева—Яо.*

Рассмотрим реализацию SMT-протокола (Secure Message Transmission), принадлежащего к группе 0-секретных протоколов передачи сообщений [1], в обобщенном канале связи.

Под обобщенным каналом связи понимается ориентированный граф с множеством вершин и путей (каналов), таких что:

1) все пути C_i $(i = 1, ..., n)$ попарно не пересекаются, за исключением оконечных точек A и B, задающих направление передачи данных в канале связи;

2) $\Pi(A, B, C, S)$ — протокол обмена сообщениями между вершинами A и B. В рамках протокола секретное сообщение (секрет) S должен быть передан в точку B по обобщенному каналу $C = \{C_1, ..., C_n\}$.

3) секрет S является элементом числового поля GF_p (p—большое простое число), изначально находящимся в точке A;

4) подчиненный модели безопасности Долева—Яо [2] протокол предусматривает противодействие противника на протяжении всей работы.

В классических вариациях SMT-протокола всегда можно выделить два этапа: разделение сообщения S на криптографические части — тени $Share_1(S)$, ..., $Share_n(S)$ с отправкой $Share_i(S)$ по каналам C_i, и сборка сообщения S из необходимого набора теней в другой точке сетевого графа.

Свойства конфиденциальности SMT-протоколов изучены в работах [1, 3, 4] для различных реализаций схем разделения и сборки секрета из теней. В частности, в статье [1] вводится понятие надежности. Надежность работы протокола $\Pi(A, B, C, S)$ определяется вероятностью P того, что по окончании работы протокола переданное и полученное сообщения одинаковы:

$$S^A = S^B, \quad P \sim 1. \tag{1}$$

В работе [4] для контроля надежности $\Pi(A, B, C, S)$ предлагается дополнительно использовать обратный канал связи (feedback).

В рамках нашей реализации протокола схема разделения секрета представляет собой классическую (n,n)-пороговую схему, позволяющую вычислять теневые копии для больших

значений n. Схема разделения секрета Шамира использует многочлены степени n–1 над полем GF_p [5]:

$$p_{n-1}(x) = a_{n-1}x^{n-1} + ... + a_1 x + a_0, \;\; a_0 = S, \tag{2}$$

$$a_{n-1}a_1 \in \text{rand}GF_p; \;\; x \in GF_p.$$

Долей секрета $Share_i(S)$ является упорядоченная пара:

$$Share_i(S) = (i, p_{n-1}(i)), \;\; i \neq 0.$$

Теорема о полиномиальной интерполяции для многочленов (2) над полем GF_p сохраняет свою силу, поэтому при наличии совокупности попарно различных теней мощности n секрет S восстанавливается по интерполяционной формуле Лагранжа:

$$p_n(x) = \prod_{i=1, j\neq i}^{n-1} \frac{x - x_i}{x_i - x_i}.$$

Обратимся к деталям реализации протокола $\Pi(A, B, C, S)$ на базе языка Python.

Для генерации и хранения теней секрета применяется тип dictionary, позволяющий использовать индексы на всем диапазоне bigint. Это позволяет фактически ограничивать количество генерируемых теней лишь объемом оперативной памяти [6] и достигать n~500 при приемлемых значениях времени работы протокола.

В протоколе используются основные математические операции над полем GF_p, реализованные в рамках языка Python, такие как сложение, умножение Карацубы и быстрое возведение в степень. Анализ быстродействия данных операций на числах различной длины позволил определить оптимальные по скорости выполнения реализации. Для качественной генерации случайных коэффициентов в выражении (2) использован алгоритм Mersenne twister (MT19937), его период равен 2^{19937}–1. Алгоритм обладает достаточно высокими показателями времени генерации, например, время генерации a_i $(i = n-1, ..., 1)$ в (2) относительно линейных конгруэнтных генераторов меньше приблизительно в 2—2,5 раза.

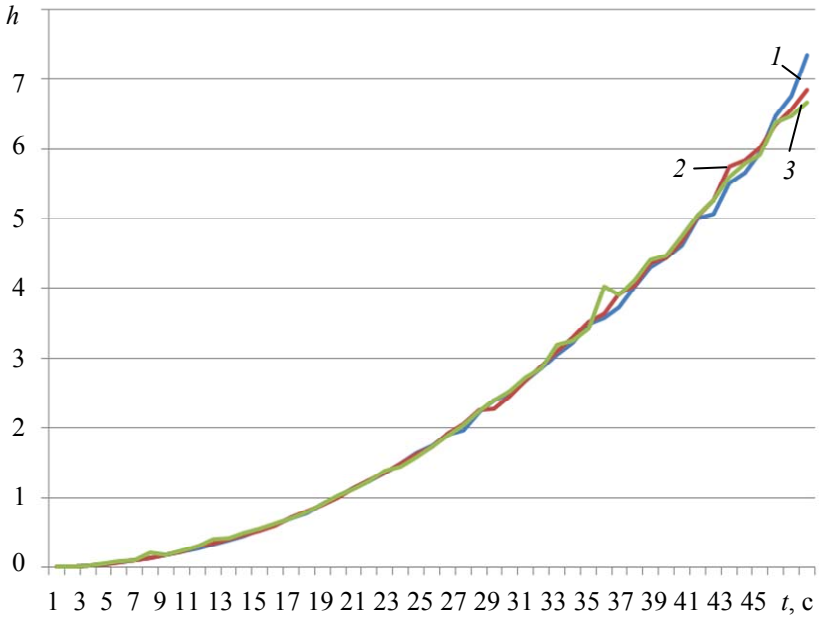
Функциональность реализованного базового SMT-протокола фактически ограничена предоставленными ресурсами оперативной памяти для хранения теней и элементов поля GF_p в процессе вычислений, а также вычислительными ресурсами процессора.

Поскольку основная часть протокола сводится к операциям разделения и сборки секретного сообщения, основные показатели быстродействия алгоритма в целом зависят от этих операций. Теоретические оценки вычислительной сложности операций разделения и сборки в нашем случае — $O(\log_2 n)$ и $O(n^2)$ соответственно.

Рассмотрим надежность работы протокола. Введение обратного канала для контроля надежности может существенно понижать конфиденциальные свойства протокола [4]. Поэтому нами предложено использовать хеш-функции как для контроля целостности теней секрета, так и для контроля сборки значения S в точке B. В частности, протокол $\Pi(A, B, C, S)$ завершает свою работу пересылкой значения хеш-функции секрета $h(S)$ по всем доступным каналам C_i либо пересылкой значений секрета и тени S+$Share_i$ по соответствующим каналам передачи теней C_i. При этом, очевидно, вероятность P в (1) зависит от криптографических свойств применяемой хеш-функции:

$$P\left[\left(h(S^A) = h(S^B)\right) \rightarrow \left(S^A = S^B\right)\right] = 1 - P(h).$$

В правой части равенства $P(h)$ — вероятность появления коллизии для выбранной криптографической хеш-функции $h(S)$.



Для вариации протокола получены значения времени (см. рисунок, кривая *1*; кривая *2* — $h(S_i)$, *3* — $h(S)$) и алгоритмической сложности сборки секрета в зависимости от значения n, соответствующие указанным выше теоретическим оценкам, представленны на рисунке.

СПИСОК ЛИТЕРАТУРЫ

1. *Dolev D., Dwork C.* Perfectly Secure Message Transmission // Proc. 31st Annu. Symp. on Found. of Comput. Sci. 1990. P. 36—45.

2. *Dolev D., Yao A.* On the Security of Public Key Protocols // IEEE Transact. on Inform. Theory. 1983. Vol. 29, N 2. P. 198—208.

3. *Kurosawa K.* General Error Decodable Secret Sharing Scheme and Its Application. Cryptology ePrint Archive Report/ 2009. P. 263.

4. *Yang Q., Desmedt Y.* Cryptanalysis of Secure Message Transmission Protocols with Feedback // ICITS. 2009. P. 159—176.

5. *Shamir A.* How to share a secret // Communication of ACM. 1979. Vol. 22, N 11. P. 612—613.

6. Python Core Development [Электронный ресурс]: <http://www.python.org/dev/peps/pep-0237/>.

Сведения об авторах

Максим Васильевич Белоусов — аспирант; Владимирский государственный университет им. А. Г. и Н. Г. Столетовых, кафедра информатики и защиты информации; E-mail: lib.bmw@gmail.com

Алексей Викторович Александров — канд. физ.-мат. наук, доцент; Владимирский государственный университет им. А. Г. и Н. Г. Столетовых, кафедра информатики и защиты информации; E-mail: alex_izi@mail.ru

Рекомендована ВлГУ

Поступила в редакцию
17.04.12 г.