ОСНОВЫ ПОСТРОЕНИЯ СИСТЕМЫ КРИПТОГРАФИЧЕСКОЙ ГЕНЕРАЦИИ ИДЕНТИЧНОЙ ИНФОРМАЦИИ

Елисеев С.О., Крюков Д.А.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский Технологический университет» (МТУ), 119454, Россия, г. Москва, проспект Вернадского, д. 78, e-mail: ideawade@gmail.com, dm.bk@bk.ru

В статье рассматриваются основные подходы, направленные на решение задачи построения и реализации системы криптографической генерации идентичной информации от отправителя к получателю. В основе предложенной методики лежит фундаментальный криптографический алгоритм Диффи-Хеллмана, предназначенный для получения общего секрета и методика формирования идентичных фрагментов данных на стороне получателя. Предложено алгоритмическое обеспечение, повышающее эффективность и производительность работы системы, проведена предварительная оценка скорости генерации на реальных данных, рассмотрены меры противодействия атакам типа «человек-посередине»

Ключевые слова: алгоритм Диффи-Хеллмана, передача данных, суффиксный массив, криптография.

FUNDAMENTALS THE CRYPTOGRAPHIC GENERATION SYSTEM OF IDENTICAL INFORMATION CONSTRUCTION

S. Eliseev, D. Kryukov

Federal State Educational Institution of Higher Education "Moscow Technological University", 119454, Russia, Moscow, Vernadscogo avenue, 78 e-mail: ideawade@gmail.com,: dm.bk@bk.ru

The paper considers the main approaches aimed at solving the problem of constructing and implementing a cryptographic generation system of identical information from the sender to the recipient. The proposed methodology is based on the fundamental cryptographic Diffie-Hellman algorithm, designed to obtain a common secret and the method of the client side forming identical data fragments. Algorithmic support is proposed that increases the efficiency and total system performance. Preliminary generation rate estimation with the real instances and measures of counteraction to MITM-attacks are considered.

Key words: Diffie-Hellman algorithm, data transfer, suffix array, cryptography.

Открытое распространение вспомогательных элементов, безопасно образующих общий секрет на основании неразрешимости задачи дискретного логарифмирования, позволяет паре различных пользователей системы выработать единую последовательность символов в незащищенном канале связи. Эффективность данного подхода обуславливается доказанной криптостойкостью такой системы и основана на высокой вычислительной сложности обращения показательной функции (1). Она вычисляется достаточно эффективно, в то время как даже самые современные алгоритмы решения задачи дискретного логарифмирования (2) являются малоэффективными. При большом значении "a" нахождение решения для такого уравнения потребует значительных временных ресурсов и сопоставимо с решением задачи полного перебора.

$$f(x) = a^x \quad (1)$$

$$A = g^a mod\ p \quad (2)$$

Следовательно, система, использующая подобный метод шифрования, будет обладать высокой криптостойкостью. Данные соотношения, заданные односторонними функциями, лежат в основе ассиметричной криптографии. Вместе с тем, их особенности позволяют разработать способ защищенной передачи данных, свободный от таких абстракций, как открытый или закрытый ключи[1].

Метод криптографической генерации данных можно частично сравнить с end-to-end шифрованием. Фактически, такой метод исключает возможность прослушивания канала кем-либо. Основной особенностью и отличием описываемого метода от end-to-end шифрования, является то, что в традиционном подходе, ключи шифрования хранятся на устройстве пользователя[1]. В случае с безопасной генерацией информации все

данные для шифрования так же хранятся исключительно на стороне клиентов, однако, они не участвуют в алгоритмических преобразованиях над открытыми данными непосредственно. Поэтому, можно с уверенностью утверждать, что злоумышленник, имея доступ в любой форме к устройству не сможет ознакомиться с передаваемыми сообщениями.

В статье предлагается рассмотреть алгоритм криптографической генерации идентичных данных в информационных системах. Описываемый алгоритм генерации данных, имеет в своей основе фундаментальный алгоритм Диффи-Хеллмана, с помощью которого возможно образовать общие для участников информационного обмена секретные ключи для алгоритма шифрования. Стоит отметить, что особенностью создаваемого алгоритма является тот факт, что в нем не предусматривается прямая передача зашифрованного текста, обмен происходит только вычислениями ассиметричной криптографии, а так же вспомогательными параметрами, не имеющими определяющего значения для потенциального атакующего, которые будут использоваться для восстановления текста на приемной стороне, и, следовательно, принимать полученные с помощью алгоритма Диффи-Хеллмана числа, в качестве ключей шифрования в данном случае не объективно.

Итак, алгоритм получения общих ключевых чисел (3) выглядит следующим образом: предположим, что с системой работают 2 пользователя Алиса и Боб. Алиса генерирует свое открытое число: $A = g^a mod\ p$, Боб генерирует свое: $B = g^b mod\ p$, числа a и b пользователи генерируют самостоятельно. Далее происходит обмен полученными значениями, и оба пользователя получают общее секретное число K.

$$K = B^a mod\ p \qquad (3)$$

$$K = A^b mod\ p$$

Как было сказано выше, полученное значение K, мы не рассматриваем, как ключ шифрования, а рассматриваем как идентичную на обеих сторонах последовательность байтов. Очевидно, что последовательность байтов, полученная подобным образом, эквивалентная для обеих сторон информационного обмена, не будет представлять собой осмысленный текст, но в этой последовательности могут находиться n случайных подпоследовательностей длиной $S_1, S_2, \dots, S_{n-1}, S_n$ байт, которые интересуют нас с точки зрения алгоритма генерации текста. Таким образом, назовем последовательность K «ключевой последовательностью» (КП), а ее подпоследовательности «текстовыми фрагментами» (ТФ).

Итак, на стороне Боба необходимо воспроизвести сообщение Алисы без непосредственной его передачи. Учитывая то обстоятельство, что у Алисы и Боба имеется идентичная последовательность байт, полагается целесообразным пошагово выполнять сравнение КП с текстом, который Алиса намерена воспроизвести у Боба. Сравнение производится целью, поиска общих ТФ, входящих в КП и в сообщение, которые можно использовать в информационном обмене неявно, то есть сведения, о которых можно передавать от Алисы к Бобу. Для повышения эффективности данной процедуры предлагается использовать существующие алгоритмы поиска подпоследовательностей в тексте неоднократно доказавшие свою эффективность в различных реализациях.

На сегодняшний день существуют множество способов нахождения произвольного набора символов в исходном тексте, такие как, например, метод построения суффиксного дерева, алгоритм Рабина-Карпа, алгоритм Кнута-Морриса-Пратта[2].

Для реализации предлагается рассмотреть построение суффиксного массива. Данный алгоритм имеет ряд преимуществ, связанных с оптимизированным процессом поиска без серьезных затрат оперативной памяти.

Суффиксные массивы используются в тех ситуациях, в которых необходимо быстро выявить подстроки, как, например, в нашем случае. Алгоритм построения суффиксного массива и будет использоваться для сравнения частей ключевой последовательности и исходного текста.

Используя суффиксный массив, можно перебирать входной текст неограниченное число раз, начиная с перебора по одному символу, смещаясь каждую итерацию на один символ, и заканчивая в тот момент, когда подстроки заданной длины найдены не будут, что следует квалифицировать как деградацию КП. В тот момент, когда совпадения будут найдены, необходимо запомнить индекс ТФ в КП (его смещение) и длину самого ТФ, а так же индекс ТФ в исходном тексте. Эти численные параметры будут использоваться для передачи собеседнику и восстановления текста. То есть, при получении таких параметров от Алисы, Боб, зная смещение и длину ТФ, сможет найти в своей, общей с Алисой ключевой последовательности, необходимый ТФ и использовать его в процедуре восстановления текста.

После деградации последовательности (при отсутствии совпадений текста и ключевой последовательности), появится необходимость сообщить об этом принимающей стороне. В качестве информирования второй стороны предлагается транслировать три параметра, необходимых для

инициализации алгоритма Диффи-Хеллмана и служащие для генерации новой ключевой последовательности. Таким образом, ключевая последовательность может подлежать ротации несколько раз за один сеанс связи, что повышает безопасность системы в целом.

Значения можно передавать в открытом канале, без применения средств криптографии. Разумеется, злоумышленник, пассивно прослушивающий канал, будет иметь возможность получить все значения в рамках информационного обмена, однако, не имея ключевой последовательности можно равновероятно ассоциировать значения с символами сообщения, что окажется существенным препятствием, таким образом, совокупность данных значений не несут смысловой нагрузки для атакующего.

Для передачи параметров восстановления текста и параметров для Алгоритма Диффи-Хеллмана по сети предлагается использовать формат JSON.

JSON (JavaScript Object Notation) - простой формат обмена данными, удобный для чтения и написания как человеком, так и компьютером. Он основан на подмножестве языка программирования JavaScript, определенного в стандарте ECMA-262 7th Edition[3]. Используя такой формат взаимодействия между клиентами, можно говорить о кроссплатформенном характере системы, возможности ее реализации на любых устройствах и на любом языке.

После того, как обеими сторонами получена новая общая последовательность, отправляющей стороне необходимо снова воспользоваться методом, поиска ТФ для генерации параметров для передачи. В конечном итоге, для всего текста сгенерируются параметры для восстановления.

Предварительные исследования показали, что скорость работы системы по создаваемому протоколу в целом не зависит от длины передаваемого сообщения. Для того, чтобы сгенерировать параметры передачи и восстановить по ним сообщение, в среднем требуется 500 миллисекунд. Исследования проводились для сообщений длиной от 10 до 1000 символов длины. Связано это с тем, что любой алфавит конечен, и, соответственно, время генерации параметров и восстановление сообщения будет упираться во время обработки всех букв алфавита того языка, на котором мы передаем сообщение – в настоящем исследовании сообщения передаются на английском языке. Время генерации и восстановления колеблется в диапазоне 150 миллисекунд от среднего значения, так как КП получается относительно случайно, что невозможно предугадать будет ли в новой последовательности необходимые нам символы или нет. Усредненные результаты исследования сведены в график, представленные на рисунке 1.

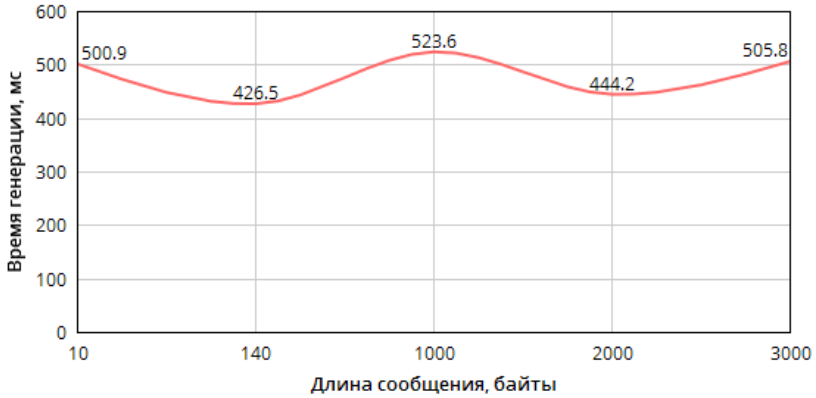

Рисунок 1. График среднего времени генерации сообщений разной длины.

Как было сказано ранее, алгоритм Диффи-Хеллмана, используемый в системе для получения КП, устойчив к атаке прямого перебора, но не может обеспечить защиту от активного прослушивания канала связи, то есть возможности внесения изменений (подлога) в информационный обмен сторон (т.н. атаки «человек-посередине»)[4]. Для обеспечения безопасности пользователей от атаки «человек посередине» предлагается два варианта. Традиционный подход, связанные с подписыванием компонентов образования общей ключевой последовательности сертификатами открытого ключа формата X.509[5][6]. При этом следует принимать во внимание необходимость существования центра сертификации, которому «доверяют» обе стороны. Второй вариант заключается в использовании дополнительного канала связи между собеседниками. Оба клиента должны быть подключены к серверу авторизации. Во время получения КП оба собеседника отправляют на сервер сокращенный результат применения хэш-функции от КП.

Серверу следует сравнить между собой полученные данные и отправлять пользователям положительный ответ, если хэш-значения совпали или отрицательный ответ, если хэш-значения не совпадают. Если хэш-

значения различны, то существует вероятность того, что канал прослушивается и в механизм образования КП были внесены подложные значения прослушивающим субъектом.

Таким образом, в статье рассмотрены основные подходы к построению и реализации системы крипографической генерации идентичной информации. В основе предложенного метода безопасной передачи данных использованы однонаправленные функции фундаментального криптографического алгоритма Диффи-Хеллмана, предназначенные для получения общего секрета. Проведена предварительная оценка скорости работы системы, показавшая, что в реализованной системе скорость передачи данных в целом не зависит от длины передаваемого сообщения. Были предложены потенциальные методы противодействия атакам типа «человек посередине».

Список литературы

1. Greenberg A. Hacker Lexicon: What Is End-to-End Encryption. // Wired интернет-журнал 25.11.14. URL: https://www.wired.com/2014/11/hacker-lexicon-end-to-end-encryption/ (дата обращения: 18.11.2017).
1. Charras C., Lecroq T. Exact string matching algorithms // Алгоритмы работы с подстроками: сайт. — URL: www-igm.univ-mlv.fr/~lecroq/string/index.html (дата обращения: 10.09.2017).
2. Стандарт ECMA-262, 8-е издание, 2017. URL: www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf (дата обращения: 11.09.2017).
3. Schneier B. Applied Cryptography: Protocols, Algorithms and Source Code in C. / WILEY, 2015. – 675 с.
4. Callegati F., Cerroni W., Ramilli M. Man-in-the-Middle Attack to the HTTPS Protocol // IEEE Security & Privacy. – 2009. – №7. – С. 78 – 81 DOI: 10.1109/MSP.2009.12.
5. Turcotte Y. Syntax testing of the entrust public key infrastructure for security vulnerabilities in the X.509 certificate, диссертация // Royal Military College of Canada (Canada). — 2005.

References

1. Greenberg A. Hacker Lexicon: What Is End-to-End Encryption. // Wired 25.11.14, Available at: https://www.wired.com/2014/11/hacker-lexicon-end-to-end-encryption/ (accessed: 18.11.2017).
2. Charras C., Lecroq T. Exact string matching algorithms // Exact string matching algorithms, Available at: URL: www-igm.univ-mlv.fr/~lecroq/string/index.html (accessed: 10.09.2017).
3. Standard ECMA-262, 8-th edition, 2017, Available at: www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf (accessed: 11.09.2017).
4. Schneier B. Applied Cryptography: Protocols, Algorithms and Source Code in C. / WILEY, 2015. – 675 p.
5. Callegati F., Cerroni W., Ramilli M. Man-in-the-Middle Attack to the HTTPS Protocol // IEEE Security & Privacy. – 2009. – Vol. 7. – pp. 78 – 81 DOI: 10.1109/MSP.2009.12.
6. Turcotte Y. Syntax testing of the entrust public key infrastructure for security vulnerabilities in the X.509 certificate // Royal Military College of Canada (Canada). — 2005.