Т. А. Билык, А. Ю. Нестеренко

КОД АУТЕНТИФИКАЦИИ СООБЩЕНИЙ НА ОСНОВЕ УНИВЕРСАЛЬНОГО ХЭШИРУЮЩЕГО ПРЕОБРАЗОВАНИЯ

Введение

Коды аутентификации сообщений являются эффективным средством подтверждения подлинности и целостности данных при их хранении и/или передаче по каналам связи, широко применяемым в криптографических протоколах (например, SSL/TLS) и различных прикладных системах.

В общем случае отправитель на основе сообщения и секретного ключа, известного отправителю и получателю сообщения, вычисляет значение кода аутентификации, называемое имитовставкой. Далее сообщение с имитовставкой пересылается получателю. Получатель, в свою очередь, заново вычисляет значение имитовставки и сверяет результат с имитовставкой, полученной от отправителя. В случае совпадения сообщение считается подлинным.

Таким образом, код аутентификации сообщений — это функция $h : M \times K \to Y$, получающая на входе сообщение $m \in M$ и секретный ключ $k \in K$ и выдающая на выходе имитовставку $y \in Y$. Такую функцию h мы можем рассматривать как ключевую функцию хэширования. Как будет показано далее, важнейшими параметрами, определяющими стойкость алгоритма, являются длина секретного ключа (обозначим её t) и длина имитовставки (обозначим её n), измеряемые в битах.

1. Постановка задачи

В настоящее время в России стоит проблема разработки алгоритма кода аутентификации сообщений. Существующий алгоритм ГОСТ 28147-89 в режиме выработки имитовставки, являющийся алгоритмом кода аутентификации, на данный момент не удовлетворяет требованиям безопасности, так как длина формируемых имитовставок составляет 32 бита и злоумышленник за короткий период времени может реализовать атаку полного перебора. Помимо данного алгоритма в России нет стандартов кодов аутентификации. Поэтому необходимо сформулировать требования к коду аутентификации сообщений, а также разработать удовлетворяющий им алгоритм.

2. Требования к коду аутентификации сообщений

Требования к кодам аутентификации сообщений приводились в [2] и заключаются в следующем.

2.1. Скорость работы

Разрабатываемый алгоритм должен иметь высокую скорость вычислений и эффективную реализацию. При этом скорость его работы должна быть сопоставима со скоростью работы безключевых алгоритмов хэширования.

2.2. Гибкость

Алгоритм должен быть универсальным и настраиваемым, чтобы удовлетворять эксплуатационным требованиям различных систем, где он внедряется, по следующим параметрам:

1) максимально допустимый объем памяти, занимаемый ключевой информацией (определяется из доступного объема памяти для ключевого носителя и пропускной способности каналов, используемых для распространения этой информации);

2) максимально допустимый объем памяти, занимаемый имитовставкой (определяется из доступного объема памяти, где предполагается хранить имитовставки с учетом максимально возможного объема хранения данных, и/или пропускной способности каналов, по которым будут передаваться эти имитовставки с учетом максимальной нагрузки на каналы для передачи самих сообщений);

3) время актуальности защищаемой информации, на протяжении которого алгоритм должен быть стоек к атакам злоумышленников (с учетом срока действия секретного ключа).

2.3. Стойкость к атакам

Атаки на коды аутентификации сообщений и оценки сложности их реализации обсуждались в [1], приведем основные положения.

2.3.1. Стойкость к атакам на ключ

Задача подбора верного значения ключа за время его действия должна быть трудноразрешимой (т. е. задача должна быть не решаема с использованием современной вычислительной техники за приемлемое время). При этом сложность подбора ключа должна оцениваться величиной $O(2^t)$.

2.3.2. Стойкость к атакам подбора имитовставки

Задача подбора верного значения имитовставки для данного сообщения m при неизвестном секретном ключе k должна быть трудноразрешимой, сложность подбора имитовставки должна оцениваться величиной $O(2^n)$.

2.3.3. Стойкость к нахождению коллизии

Алгоритм должен удовлетворять следующим требованиям:

1) При неизвестном секретном ключе k и известном сообщении m задача подбора сообщения $m_1 \neq m$, такого, что $h(k, m_1) = h(k, m)$, должна быть трудноразрешимой и оцениваться величиной порядка $O(2^{n/2})$.

2) При неизвестном секретном ключе k задача выбора сообщений m_1, m_2, таких, что $m_1 \neq m_2$ и $h(k, m_1) = h(k, m_2)$, должна быть трудноразрешимой как при известном, так и при неизвестном значении $h(k, m_1)$.

С учетом парадокса задачи о днях рождения сложность нахождения коллизии оценивается как $O(2^{n/2})$.

Учитывая современные вычислительные возможности, сложность перечисленных в п. 2.3.1-2.3.3 атак должна оцениваться величиной $O(2^{128})$, тогда значения длины имитовставки n и длины ключа t должны удовлетворять соотношению:

$$\begin{cases} t \geq 128 \\ n \geq 256 \end{cases}.$$

3. Разработка кода аутентификации сообщений

3.1. Выбор архитектуры кода аутентификации сообщений

Существуют три основных подхода построения кода аутентификации сообщений:

1) на основе блочного шифрования (например, алгоритм CMAC [3]);

2) на основе хэш-функции (например, алгоритм HMAC [4]);

3) на основе универсального класса хэш-функций (например, алгоритм UMAC [5]).

Основываясь на анализе существующих зарубежных алгоритмов, мы считаем, что наиболее перспективным с точки указанных ранее требований будет использование третьей архитектуры в качестве основы нового алгоритма.

Последний подход берет свое начало с работ Л. Картера и М. Вегмана [6, 7], в которых было введено понятие универсального класса хэш-функции и предложено его использование для построения кода аутентификации.

Универсальный класс хэш-функций — множество функций, обладающих определенными комбинаторными свойствами. Для вычисления кода аутентификации Л. Картер и М. Вегман предложили применять к входному сообщению случайно выбранную функцию из строго универсального класса хэш-функций и результат шифровать наложением одноразовой гаммы.

Позже этот подход построения кодов аутентификации был развит в работах [8, 9] для создания алгоритма UMAC, где к сообщению применяются функция из универсального класса хэш-функций (выбор функции определяет секретный ключ), реализующая трехэтапное сжатие сообщения, и наложение выхода псевдослучайной функции.

Благодаря тому, что финальное шифрование применяется к уже сжатому хэш-функцией сообщению, скорость работы алгоритма гораздо выше, чем у кодов аутентификации, построенных по первой и второй архитектурам.

Однако UMAC имеет ряд минусов: он вычисляет имитовставки длиной не более 128 бит с использованием в качестве блочного шифрующего преобразования алгоритм AES с ключами длиной 128, 192 либо 256 бит. Таким образом, данный алгоритм удовлетворяет указанным в п. 2.3 требованиям только с параметрами максимальной длины и не позволяет повышать уровень защищенности, увеличивая длину ключа и длину имитовставки. Данные минусы устранены в предложенном нами алгоритме.

3.2. Описание нового алгоритма кода аутентификации сообщений

Разработанный нами алгоритм кода аутентификации сообщений использует псевдослучайную функцию PRF и алгоритм перестановки, описание которых дадим отдельно.

3.2.1. Функция PRF

В качестве функции PRF может использоваться генератор псевдослучайных чисел, который по заданному начальному значению K вырабатывает последовательность чисел $k_0, k_1, \ldots$. Помимо этого, генератор должен удовлетворять следующим требованиям:

1) выход функции при неизвестном ключе должен быть непредсказуем;

2) последовательность чисел $k_0, k_1, \ldots$ должна быть статистически неотличима от последовательности равномерно распределенных на интервале $[0; 2^w - 1]$ чисел.

3) по выходу функции должно быть сложно определить ключ K.

В качестве оптимального решения функция PRF может быть заменена на поточный шифр, удовлетворяющий перечисленным требованиям. Вместе с тем подробное описание поточного шифра лежит за рамками данной работы и в статье далее не рассматривается.

3.2.2. Выработка перестановки π

На вход алгоритм получает ключ K и параметр u, определяющий число элементов, задающих перестановку. Для выработки перестановки π вычисляем последовательность элементов и $\varepsilon_j \neq \varepsilon_k$ для $\forall k \neq j$ как очередной выход функции PRF(K). Для этого на каждом шаге генерации элемента ε_j проверяем, что $\varepsilon_k \neq \varepsilon_l$ для $k < j$, если повтор встретился, то ε_j отбрасывается и вырабатывается новый элемент.

На выходе получаем $\pi = (\varepsilon_0, \ldots, \varepsilon_{u-1})$.

3.2.3. Основной алгоритм кода аутентификации сообщений

Вход: n — длина имитовставки в битах, n кратно 8 и $n \geq 256$.

K — секретный ключ длиной t бит, равной длине ключа, используемого в функции PRF, $t \geq 128$.

M — сообщение произвольной длины.

Выход: tag — имитовставка длиной n бит.

1) Обозначим $w = \frac{n}{2}$ — определяет размер в битах блоков разбиения входного сообщения, $u = \frac{n}{8}$ — длина имитовставки в байтах, $lenM$ — длина сообщения M в битах.

2) Дополняем входное сообщение последовательностью нулей и значением $lenM$ таким образом, чтобы результат $M' = M \,||0..0||\, lenM$ имел длину $lenM'$, кратную 2w.

3) Представим M' в виде последовательности пар чисел $(m_0, m_1), \ldots, (m_{S-2}, m_{S-1}) \in \mathbf{Z}_{2^w}$, где $s = \frac{lenM'}{w}$.

4) Вычисляем последовательность рабочих ключей $k_0, \dots, k_{S-1}$, где $k_i \in \mathbf{Z}_2 w, i = \overline{0, s-1}$ как выход псевдослучайной функции PRF(K).

5) Вычисляем

$$H = \sum_{i=0}^{\frac{S}{2}-1} (k_{2i} + m_{2i} \ (mod\ 2^w)) \cdot (k_{2i+1} + m_{2i+1} \ (mod\ 2^w)) (mod\ 2^{2w}) \ . \quad (1)$$

6) Представляем $H \in \mathbf{Z}_2 2^w$ в виде последовательности байтов

$H = \{h_0, \dots, h_{u-1}\}$, где $h_i \in \mathbf{Z}_2 8$ для $i = \overline{0, u-1}$ и $H = \sum_{i=0}^{u-1} h_i \cdot (2^8)^i$.

7) Вырабатываем перестановку $\pi = (\varepsilon_0, \dots, \varepsilon_{u-1})$.

8) Выполняем финальную перестановку байтов полученной последовательности H:

Цикл по i от 0 до (u – 1)

$h_i' = h_{\varepsilon_i}$.

9) $Tag = \{h_0', \dots, h_{u-1}'\}$.

4. Анализ предложенного алгоритма

4.1. Анализ преобразования H

Свойства преобразования (1) изучались в работе [9]

Теорема. Зафиксируем натуральное четное число $S > 1$ и произвольное натуральное число w.

Пусть M — конечная последовательность целых неотрицательных чисел $m_0, \dots, m_{S-1} \in \mathbf{Z}_2 w$, K — конечная последовательность целых неотрицательных чисел $k_0, \dots, k_{S-1} \in \mathbf{Z}_2 w$. Определим функцию двух переменных

$$H(M, K) = \sum_{i=0}^{\frac{S}{2}-1} (k_{2i} + m_{2i} \ (mod\ 2^w)) \cdot (k_{2i+1} + m_{2i+1} \ (mod\ 2^w)) (mod\ 2^{2w}).$$

Тогда функция H является универсальной функцией хэширования со значением $\varepsilon = \frac{1}{2^w}$, т. е. для двух различных сообщений M и M' вероятность совпадения значений функции H не превосходит величины $P(H(M) = H(M')) \le \frac{1}{2^w}$.

Доказательство

1) Зафиксируем некоторый индекс i и рассмотрим сообщение

$$M' = m_0, \dots, m_{2i-1}, m_{2i}', m_{2i+1}', m_{2i+1}, \dots, m_{S-1},$$

отличающееся от сообщения M только значениями величин m_{2i}', m_{2i+1}'.

2) Фиксируем значение ключа K и обозначим

$$A = (k_{2i} + m_{2i}' \ (mod\ 2^w)) \cdot (k_{2i+1} + m_{2i+1}' \ (mod\ 2^w)) (mod\ 2^{2w}).$$

При любом значении A данное уравнение имеет не более 2^w решений относительно неизвестных m_{2i}', m_{2i+1}'. Так как при фиксированном i мощность множества возможных пар (m_{2i}', m_{2i+1}') равняется 2^{2w}, получаем, что при случайном выборе неизвестных m_{2i}', m_{2i+1}' вероятность появления коллизии $(k_{2i} + m_{2i} \ (mod\ 2^w)) \cdot (k_{2i+1} + m_{2i+1} \ (mod\ 2^w)) (mod\ 2^{2w}) = (k_{2i} + m_{2i}' \ (mod\ 2^w)) \cdot \left(k_{2i+1} + m_{2i+1}' \ (mod\ 2^w)\right) (mod\ 2^{2w})$ равна и не зависит от i

Обозначим $H(K, M) = \sum_{i=0}^{S-1} h_i \ (mod\ 2^{2w})$, $H(K, M') = \sum_{i=0}^{S-1} h_i' \ (mod\ 2^{2w})$. Выше мы оценили вероятность появления коллизии, когда $h_i = h_i'$ для некоторого i, т. е. случай совпадения одного слагаемого. Теперь найдем вероятность возникновения коллизии, когда $h_i + h_j = h_i' + h_j' (mod 2^{2w})$, т. е. коллизии, образованной двумя слагаемыми. При фиксированных h_i, h_j указанное сравнение имеет 2^{2w} решения относительно неизвестных h_i', h_j'. Так как мощность множества возможных пар (h_i', h_j') равняется 2^{4w}, получаем, что вероятность коллизии равна $\frac{1}{2^{2w}}$, что удовлетворяет условию теоремы.

Рассмотрение случаев для большего числа слагаемых $h_1, h_2, \ldots$ сводится к случаю двух слагаемых. Теорема доказана.

5.2. Анализ финальной перестановки

Преобразования, выполняемые в пунктах 7 и 8 алгоритма, являются обязательными, в противном случае существует эффективный алгоритм восстановления последовательности рабочих ключей.

Пусть последнее преобразование пункта 7 не выполняется и злоумышленник перехватил две пары сообщение-имитовставка: $(m_{1,0}||m_{1,1}, Tag_1)$, $(m_{2,0}||m_{2,1}, Tag_2)$. Тогда злоумышленник может составить систему уравнений от двух неизвестных k_0, k_1 :

$$\begin{cases}(k_0 + m_{1,0})(k_1 + m_{1,1}) = Tag_1 (mod\ 2^{2w}) \\ (k_0 + m_{2,0})(k_1 + m_{2,1}) = Tag_2 (mod\ 2^{2w})\end{cases}.$$

Из данной системы уравнений однозначно определяются k_0, k_1. Алгоритм восстановления последовательности рабочих ключей для большего числа сообщений аналогичен.

Заключение

Разработан новый алгоритм кода аутентификации сообщений, удовлетворяющий современным требованиям безопасности. Анализ предложенного алгоритма показал его стойкость к атакам злоумышленников. В дальнейшем предполагается провести тестирование скорости работы программной реализации данного алгоритма, а также статистический анализ его выходов.

СПИСОК ЛИТЕРАТУРЫ:

1) *Билык Т. А.* Атаки на коды аутентификации сообщений // Труды IX Международной научно-технической конференции «Новые информационные технологии и системы». Пенза: ПГУ, 2010. Ч. 2. С. 100–103.

2) *Билык Т. А.* Выработка требований к ключевой функции хэширования и разработка удовлетворяющего им алгоритма // Материалы Ежегодной научно-технической конференции студентов, аспирантов и молодых специалистов МИЭМ. М.: МИЭМ, 2010. С. 30–31.

3) *Song Jh., Poovendran R., Lee J., Iwata T.* The AES-CMAC Algorithm. RFC 4493. 2006. URL: http://www.ietf.org/rfc/rfc4493.txt.

4) *Krawczyk H., Bellare M., Canetti R.* HMAC: Keyed-Hashing for Message Authentication. RFC 2104. 1997. URL: http://tools.ietf.org/html/rfc2104.

5) *Krowetz T.* UMAC: Message Authentication Code using Universal Hashing. RFC 4418. 2006. URL: http://www.ietf.org/rfc/rfc4418.txt.

6) *Wegman M., Carter L.* Universal classes of hash functions // Journal of Computer and System Sciences. 1979. № 18. С. 143–154.

7) *Wegman M., Carter L.* New hash functions and their use in authentication and set equality // Journal of Computer and System Sciences. 1981. № 22. С. 265–279.

8) *Black J., Halevi S., Krawczyk H., Krovetz T., Rogaway P.* UMAC: Fast and provably secure message authentication // Advances in Cryptology — CRYPTO'99. 1999. С. 216–233.

9) *Krovetz T.* Software-optimized universal hashing and message authentication // UMI Dissertation Services. 2000. С. 34–43.