О повышении криптостойкости однонаправленных хеш-функций

В. Ю. ЛЁВИН
Московский государственный университет им. М. В. Ломоносова
e-mail: levval@yandex.ru

УДК 519.95

Ключевые слова: однонаправленные хеш-функции, цифровая подпись, криптографическая безопасность, целостность данных, идентификация цифровых сообщений.

Аннотация

В статье приводятся конструктивные предложения для решения задачи обеспечения подлинности и достоверности цифровых документов с использованием однонаправленных хеш-функций. Численно оценивается стойкость однонаправленных хеш-функций при различных видах их взлома. Предложен ряд алгоритмов, позволяющих серьёзно повысить криптостойкость хеш-функций без переделки их внутренних алгоритмов, и выбран лучший по скорости и качеству. Показано, что метод суффиксной суперпозиции Б. Шнайера не годится для использования в этих целях. Предложенные в статье методы могут быть использованы для улучшения большинства однонаправленных хеш-функций (например, MD4, MD5, RIPEMD, SHA, ГОСТ 34 11-94).

Abstract

V. Yu. Levin, The increasing of hash functions security, Fundamentalnaya i prikladnaya matematika, vol. 15 (2009), no. 5, pp. 171—179.

In this paper, we introduce a solution of a data integrity ensuring using cryptographic one-way hash functions. The cryptographical security of such hash functions was estimated by us in detail for different kinds of attacks. We propose a several new schemes of a one-way hash functions security strengthening without reformation of its internal algorithms. Also we outline schemes with the best speed and security level. We show that the Schneier method of suffix superposition has seriously drawback. In this article, we also suggest the method of constructing collision resistant one-way hash functions from standard well-known hash functions. Therefore, the proposed schemas can be used for upgrade majority cryptographic one-way functions, such as MD4, MD5, RIPEMD, SHA, GOST 34 11-94.

Область применения хеш-функций довольно широкая, в частности, они используются для составления уникального идентификационного кода передаваемого сообщения. Подобная задача особенно актуальна в электронном документообороте с использованием технологии цифровой подписи. Каждый человек имеет уникальные отпечатки пальцев и сетчатки глаза, строение и состав ДНК; каждое сообщение имеет уникальный идентификационный код (хеш-значение), вероятность существования двух одинаковых кодов пренебрежимо мала.

Фундаментальная и прикладная математика, 2009, том 15, № 5, с. 171—179.

Рассмотрим данный вопрос подробнее. Пусть $M \in \{0,1\}^*$ — произвольное цифровое сообщение.

Определение. Функция

$$h(M)\colon \{0,1\}^* \to \{0,1\}^n, \quad n \in \mathbb{N},$$

называется односторонней хеш-функцией порядка n, если выполнены следующие условия:

1) значение функции h определено для любого цифрового сообщения $M \in \{0,1\}^*$;
2) для любого цифрового сообщения $M \in \{0,1\}^*$ функция h имеет фиксированный порядок $n \in \mathbb{N}$;
3) для любого $M \in \{0,1\}^*$ значение $h(M)$ вычисляется за полиномиальное время;
4) для любого $M_1 \in \{0,1\}^*$ вычислительно сложно найти сообщение $M_2 \in \{0,1\}^*$, такое что $M_1 \neq M_2$, $h(M_1) = h(M_2)$;
5) вычислительно невозможно (за разумное время) найти пару (M_1, M_2), $M_1 \neq M_2$, $M_i \in \{0,1\}^*$, $i = 1,2$, такую что $h(M_1) = h(M_2)$.

Заметим, что существует много различных определений хеш-функций. Определение, приведённое выше, — классическое определение хеш-функции, применяющееся в криптографии [2]. Свойства 4), 5) являются важнейшими криптографическими свойствами. Действительно, рассмотрев циклически избыточный код CRC, побитный код XOR (или ротационный код RXOR), можно установить, что ни один из них не удовлетворяют свойствам 4), 5). Поэтому использовать данные алгоритмы для решения задачи обеспечения подлинности и достоверности как криптографически хорошие хеш-функции нецелесообразно. Злоумышленнику не составит особого труда подделать исходное сообщение таким образом, чтобы получилось сообщение с тем же хеш-значением. Отметим, что задача обеспечения достоверности является одной из ключевых задач в криптографии. Мы часто сталкиваемся с подобной задачей, отправляя от своего имени распоряжение в банк, отдавая команды, высылая договоры и, наконец, скачивая файлы из Интернета.

В настоящее время существует довольно много различных хеш-функций, предложенных для решения задачи обеспечения подлинности. Описание соответствующих алгоритмов можно найти в [1, 2]. Оценим криптографическую стойкость однонаправленных хеш-функций.

Так как множество аргументов хеш-функции счётно, а значения имеют определённый фиксированный порядок, то коллизии неизбежны. Цель злоумышленника — научиться заготавливать коллизии, т. е. научиться фальсифицировать хеш-значения. Рассмотрим трудоёмкости основных необходимых злоумышленнику процедур.

Грубый взлом хеш-функций (метод простого перебора)

Предположим, что злоумышленнику известен алгоритм построения хеш-функции h, первоначальное сообщение $M \in \{0,1\}^*$ и хеш-значение $h(M)$. Требуется найти такое сообщение $N \in \{0,1\}^*$, что $h(M) = h(N)$. Для произвольных $M, N \in \{0,1\}^*$ вероятность того, что $h(M) = h(N)$, равна 2^{-n}, где n — порядок хеш-функции h. Обозначим через $P_1(k,n)$ вероятность того, что для фиксированного сообщения $M \in \{0,1\}^*$ и сообщений $N_1, \ldots, N_k \in \{0,1\}^*$ существует номер $i = 1, \ldots, k$, $k \in \mathbb{N}$, такой что $h(M) = h(N_i)$. Имеем

$$P_1(k,n) = 1 - (1 - 2^{-n})^k \approx k2^{-n}.$$

Таблица 1. Значения вероятности $P_1(k,n)$

$P_1(k,n)$	k
0,01	$\approx 2^{n-7}$
0,5	$\approx 2^{n-1}$
0,99	$\approx 2^n$

В таблице 1 представлены значения вероятности $P_1(k,n)$ и длины перебора текстов k. Как следует из данной таблицы, злоумышленнику для подделки хеш-значения порядка n фиксированного текста этим методом потребуется перебрать не менее 2^{n-7} текстов. При этом вероятность успеха будет не выше 1 %. Всё это показывает, что для взлома фиксированного значения перебор не годится. Действительно, уже для 128-битных хеш-значений (MD2, MD4, MD5, RIPEMD128, HAVAL3-4) потребуется перебрать около 2^{120} текстов. Осуществить это за обозримое время невозможно.

Взлом хеш-функций на основе парадокса дней рождений

Атака на хеш-функции на основе парадокса дней рождений является одной из самых распространённых. Рассмотрим следующую задачу: обозначим через $P_2(k,n)$ вероятность того, что на множестве из k элементов, каждый из которых может принимать 2^n значений, есть хотя бы два с одинаковыми значениями. Выведем формулу для $P_2(k,n)$. Число различных способов выбора элементов таким образом, чтобы при этом не было дублей, равно

$$2^n \cdot (2^n - 1) \cdot \ldots \cdot (2^n - k + 1) = \frac{2^n!}{(2^n - k)!}.$$

Всего возможных способов выбора элементов 2^{kn}. Следовательно,

$$P_2(k,n) = 1 - \frac{2^n!}{(2^n - k)!} \cdot 2^{-kn}.$$

Заметим, что

$$P_2(k,n) = 1 - \left(2^n \cdot (2^n-1) \cdot \ldots \cdot (2^n-k+1)\right) \cdot 2^{-kn} =$$
$$= 1 - \left(\frac{2^n-1}{2^n} \cdot \frac{2^n-2}{2^n} \cdot \ldots \cdot \frac{2^n-k+1}{2^n}\right) =$$
$$= 1 - \left(\left(1-\frac{1}{2^n}\right) \cdot \left(1-\frac{2}{2^n}\right) \cdot \ldots \cdot \left(1-\frac{k-1}{2^n}\right)\right).$$

Используя то, что $1 - x \leqslant \exp^x$, получаем

$$P_2(k,n) > 1 - \exp^{-\frac{k(k-1)}{2^n}}.$$

Таблица 2. Значения вероятности $P_2(k,n)$

$P_2(k,n)$	k
0,01	$\approx 2^{n/2-3}$
0,5	$\approx 2^{n/2}$
0,99	$\approx 2^{n/2+2}$

Как следует из таблицы 2, при $k = 2^{n/2}$ вероятность найти коллизию больше 50 %. Подобный результат называется «парадоксом дня рождения» потому, что в соответствии с приведёнными выше рассуждениями, для того чтобы вероятность совпадения дней рождения у двух человек была больше 0,5, в группе должно быть всего 23 человека. Этот результат кажется удивительным, возможно потому, что для каждого отдельного человека в группе вероятность того, что его день рождения совпадёт с днём рождения другого человека в группе, достаточно мала. Подобная атака показывает, что порядок хеш-значения должен быть более 256 бит, чтобы сделать перебор невозможным в современных условиях (2^{128}, что пока остаётся неосуществимым в реальных условиях). Однако множество 128-битных хеш-функции можно удачно атаковать и в настоящее время. Однако не стоит считать, что 256-битные хеш-функции надёжны. В соответствии с законом Мура мощность микропроцессоров увеличивается в десять раз за каждые шесть лет, поэтому криптостойкость хеш-функций — это вопрос времени.

Взлом хеш-функций за линейное время (туннельный эффект)

Данная атака основывается на принципе построения большинства хеш-функций. Действительно, большинство хеш-функций используют функции сжатия (или сдвиговую функцию). Текст M разбивается на блоки M_i (в большинстве случаев размером в 512 бит), затем включается итерационный процесс подсчёта хеш-значения $h_i = f(h_{i-1}, M_i)$, $i \in \mathbb{N}$. Так как длина раунда небольшая,

то, применяя дифференциальный анализ, можно найти такой текст ΔC, что $h(M_i + \Delta C) = h(M_i)$.

Приведём пример. Рассмотрим широко распространённую однонаправленную хеш-функцию MD4, предложенную Роном Ривестом. MD4 обладает 128-битным выходным значением и является самой быстрой хеш-функцией. Пусть [3]

$$\Delta C = (0, 2^{31}, 2^{31} - 2^{28}, 0, 0, 0, 0, 0, 0, 0, 0, 0, -2^{16}, 0, 0, 0).$$

Тогда $\mathrm{MD4}(M + \Delta C) = \mathrm{MD4}(M)$ для любого текста M. Соответствующие результаты приведены в таблице .

Таблица 3. Пример коллизии для хеш-функции MD4

M_1	4d7a9c83 56cb927a b9d5a578 57a7a5ee de748a3c dcc366b3 b683a020 3b2a5d9f c69d71b3 f9e99198 d79f805e a63bb2e8 45dd8e31 97e31fe5 2794bf08 b9e8c3e9
M_2	4d7a9c83 d6cb927a 29d5a578 57a7a5ee de748a3c dcc366b3 b683a020 3b2a5d9f c69d71b3 f9e99198 d79f805e a63bb2e8 45dc8e31 97e31fe5 2794bf08 b9e8c3e9
$\mathrm{MD4}(M_1) = \mathrm{MD4}(M_2)$	**5f5c1a0d 71b36046 1b5435da 9b0d807a**
M_1	4d7a9c83 56cb927a b9d5a578 57a7a5ee de748a3c dcc366b3 b683a020 3b2a5d9f c69d71b3 f9e99198 d79f805e a63bb2e8 45dd8e31 97e31fe5 f713c240 a7b8cf69
M_2	4d7a9c83 d6cb927a 29d5a578 57a7a5ee de748a3c dcc366b3 b683a020 3b2a5d9f c69d71b3 f9e99198 d79f805e a63bb2e8 45dc8e31 97e31fe5 f713c240 a7b8cf69
$\mathrm{MD4}(M_1) = \mathrm{MD4}(M_2)$	**e0f76122 c429c56c ebb5e256 b809793**

Другой пример использования туннельного эффекта можно привести для RIPEMD128 [1]. Данная хеш-функция представляет собой изменённый вариант MD4, использующий другие циклические сдвиги и порядок слов сообщения. Туннельный эффект в этом случае выглядит следующим образом:

$$\Delta C = (0, 0, 0, 2^{20}, 0, 0, 0, 0, 0, 0, 2^{18} + 2^{31}, 0, 0, 0, 0, 2^{31}).$$

Тогда $\mathrm{RIPEMD128}(M + \Delta C) = \mathrm{RIPEMD128}(M)$ для любого текста M.

Данная атака является самой эффективной, ведь её трудоёмкость минимальна и сводится к простым вычислениям.

Таблица 4. Пример туннельного эффекта для хеш-функции RIPEMD128

M_1	579faf8e 9ecf579 574a6aba 78413511 a2b410a4 ad2f6c9f b56202c 4d757911 bdeaae7 78bc91f2 47bc6d7d 9abdd1b1 a45d2015 817104ff 264758a8 61064ea5
M_2	579faf8e 9ecf579 574a6aba 78513511 a2b410a4 ad2f6c9f b56202c 4d757911 bdeaae7 78bc91f2 c7c06d7d 9abdd1b1 a45d2015 817104ff 264758a8 e1064ea5
RIPEMD128(M_1)=RIPEMD128(M_2)	**1fab152 1654a31b 7a33776a 9e968ba7**
M_1	579faf8e 9ecf579 574a6aba 78413511 a2b410a4 ad2f6c9f b56202c 4d757911 bdeaae7 78bc91f2 47bc6d7d 9abdd1b1 a45d2015 a0a504ff b18d58a8 e70c66b6
M_2	579faf8e 9ecf579 574a6aba 78513511 a2b410a4 ad2f6c9f b56202c 4d757911 bdeaae7 78bc91f2 c7c06d7d 9abdd1b1 a45d2015 a0a504ff b18d58a8 670c66b6
RIPEMD128(M_1)=RIPEMD128(M_2)	**1f2c159f 569b31a6 dfcaa51a 25665d24**

Рассмотрев основные атаки, можно сделать вывод о множестве недостатков в однонаправленных хеш-функциях. Использовать такие хеш-функции, как MD4, MD5, RIPEMD128, SHA1, небезопасно, надёжность остальных остаётся под вопросом. Ещё одной проблемой является то, что перечисленные хеш-функции стандартизованы и зашиты во множество служебных библиотек и программ MS Windows, Linux. Появившиеся бреши становятся серьёзной угрозой.

Предложим несколько методов по увеличению криптостойкости однонаправленных хеш-функций. Предположим, что у нас имеется произвольная однонаправленная хеш-функция h порядка n. Рассмотрим несколько методов повышения её криптостойкости.

Метод последовательной суффиксной суперпозиции

Суть данного метода заключается в следующем (см. [1]). Пусть $M \in \{0,1\}^*$ — произвольный текст. Получим хеш-значение данного текста согласно следующему правилу:

$$\bar{h}(M) = \ldots h\Big(M \parallel h\big(M \parallel h(M)\big)\Big).$$

Метод последовательной суффиксной суперпозиции увеличивает порядок хеш-функции, но не увеличивает её криптостойкости. Проиллюстрируем сделанное замечание на примере.

Таблица 5. Пример коллизии в методе последовательной суффиксной суперпозиции

M_1	4d7a9c83 56cb927a b9d5a578 57a7a5ee de748a3c dcc366b3 b683a020 3b2a5d9f c69d71b3 f9e99198 d79f805e a63bb2e8 45dd8e31 97e31fe5 2794bf08 b9e8c3e9
MD4(M_1)	**d8024c54 82a68fec a61bb37e 35a75377**
M_2	4d7a9c83 d6cb927a 29d5a578 57a7a5ee de748a3c dcc366b3 b683a020 3b2a5d9f c69d71b3 f9e99198 d79f805e a63bb2e8 45dc8e31 97e31fe5 2794bf08 b9e8c3e9
MD4(M_2)	**d8024c54 82a68fec a61bb37e 35a75377**
MD4($M_1 \parallel MD4(M_1)$)	**4fb7eb59 7b1020d3 d4429ec7 a18be02e**
MD4($M_2 \parallel$ MD4(M_2))	**4fb7eb59 7b1020d3 d4429ec7 a18be02e**

Как следует из таблицы 5, метод последовательной суффиксной суперпозиции здесь не работает. Дело в том, что значения внутренних регистров стабилизируются.

Метод последовательной префиксной суперпозиции

Суть данного метода заключается в следующем. Пусть $M \in \{0,1\}^*$ — произвольный текст. Получим хеш-значение данного текста согласно следующему правилу:

$$h(M) = \ldots h(h(h(M) \parallel M) \parallel M) \parallel M.$$

Предложенный метод последовательной префиксной суперпозиции даёт положительный эффект, он позволяет успешно противостоять атаке, основанной на туннельном эффекте. Действительно, для текста M и текста $M + \Delta C$, построенного по туннельному эффекту ΔC, справедливо $h(M) = h(M + \Delta C)$. Однако пара текстов $h(M) \parallel M$, $h(M + \Delta C) \parallel M$ отличаются уже на $\Delta D \neq \Delta C$. Следовательно, во вновь полученном тексте вид туннельного эффекта не сохранится. Используя M_1, M_2 из таблицы 6, получим следующие результаты.

Если продолжать процедуру префиксных суперпозиций, новые хеш-значения будут сильно отличатся, так как хеш-функция $h()$ обладает лавинообразным эффектом. Единственный недостаток данного метода состоит в том, что скорость

Таблица 6. Результат работы метода последовательной префиксной суперпозиции

$\mathrm{MD4}(M_1 \parallel \mathrm{MD4}(M_1))$	**14bb2693 ebd1cbd9 28c04dc4 13652941**
$\mathrm{MD4}(M_2 \parallel \mathrm{MD4}(M_2))$	**08372524 65baf1ea 841e560c fed946c4**

выработки хеш-значения будет снижаться в пропорциональное число раз. Архитектура метода последовательной префиксной суперпозиции не подразумевает распараллеливание, что делает схему менее привлекательной. Однако за счёт увеличения порядка хеш-функции криптостойкость существенно повышается.

Метод конкатенации

Суть метода конкатенации заключается в конкатенации нескольких хеш-значений в одно:

$$h(M) = h_1(M) \parallel h_2(M) \parallel \ldots \parallel h_k(M),$$

где h_i — различные хеш-функции. Например, можно построить хеш-функцию следующим образом:

$$h(M) = \mathrm{MD4}(M) \parallel \mathrm{MD5}(M) \parallel \mathrm{SHA1}(M) \parallel \mathrm{RIPEMD128}(M) \parallel \mathrm{HAVAL128}(M).$$

Данный метод не увеличивает криптостойкости хеш-функции. Действительно, порядок этой хеш-функции будет равен 128 (а не $128 \cdot 5$). Поэтому использование метода конкатенации сомнительно. Отметим, что данный метод всё же можно использовать для улучшения криптологических свойств хеш-функций.

Метод перестановок

Для повышения криптостойкости хеш-функций лучше всего подходит метод перестановок. Хеш-функция из этого метода строится по правилу

$$h(M) = h(M) \parallel h\big(\pi_1(M)\big) \parallel \ldots \parallel h\big(\pi_k(M)\big).$$

Здесь π_i, $i = 1, \ldots, k$, — произвольные перестановки текста M. Например, можно предложить использовать каскадную схему метода перестановок. Первая перестановка получена перестановкой двух половинок текста, вторая — перестановкой четвертинок текста, третья — перестановкой восьмых долей текста и т. д. Полученное хеш-значения будет хеш-функцией как конкатенация хеш-значений однонаправленной хеш-функции. Данный метод эффективно увеличивает криптостойкость за счёт серьёзного увеличения хеш-значения. Метод перестановок также эффективен против туннельного эффекта. Дело в том, что после перестановки полученный текст отличается от исходного не на туннельный эффект ΔC, что приводит к отличию хеш-значений данных текстов. Данный метод может быть эффективно распараллелен, что является важным преимуществом

для использования его на маломощных микропроцессорах. Если в задаче скорость не является серьёзным аргументом, можно использовать усиление метода перестановок следующего вида:

$$h(M) = \ldots h\Big(\pi_2\big(h\big(\pi_1(h(M) \parallel M)\big) \parallel M\big)\Big) \ldots .$$

Если нет возможности изменить алгоритм вычисления хеш-значения, то последняя схема усиления является самой эффективной.

Литература

[1] Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. — М.: Триумф, 2002.

[2] Menezes A., Oorschot O., Vanstone S. Handbook of Applied Cryptography. — CRC Press, 1996.

[3] Wang X., Feng D., Lai X., Yu H. Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD // CRYPTO 2004.