УДК 004.021

В.В. Вихман, М.А. Панков

ПОВЫШЕНИЕ СТОЙКОСТИ ХЕШ-ФУНКЦИЙ В ИНФОРМАЦИОННЫХ СИСТЕМАХ НА ОСНОВЕ АЛГОРИТМА МНОГОИТЕРАЦИОННОГО ХЕШИРОВАНИЯ С НЕСКОЛЬКИМИ МОДИФИКАТОРАМИ

Вихман В.В., Панков М.А. **Повышение стойкости хеш-функций в информационных системах на основе алгоритма многоитерационного хеширования с несколькими модификаторами.**

Аннотация. В данной работе рассматривается влияние параметров алгоритма многоитерационного хеширования с несколькими модификаторами на его криптостойкость. Обоснована актуальность применения алгоритма многоитерационного хеширования с несколькими модификаторами и необходимость исследования его параметров, приводится описание алгоритма. Стойкость хеш-функции к атакам, не зависящим от алгоритма, обусловливается ее разрядностью, т.е. фактически – количеством уникальных значений, которое способна генерировать данная хеш-функция. Для оценки стойкости алгоритма к атакам методами «грубой силы», «дней рождения» и словарным атакам алгоритм многоитерационного хеширования с несколькими модификаторами рассматривается как самостоятельная хеш-функция. Оценку стойкости алгоритма при заданном количестве итераций предлагается производить путем вычисления средней разрядности эквивалентно стойкой хеш-функции для алгоритма. Приводится описание метода оценки стойкости алгоритма. Эксперименты производятся с использованием усеченной криптостойкой хеш-функции. Приводятся результаты экспериментов, позволяющие сравнить между собой показатели стойкости алгоритма при различных значениях его параметров. Кроме того, результаты экспериментов позволяют понять, как значения тех или иных параметров, а также сочетания значений этих параметров влияют на криптостойкость алгоритма к атакам методами «грубой силы», «дней рождения» и словарным атакам. На основании полученных результатов можно сделать выводы о значениях параметров, рекомендуемых для практического применения данного алгоритма. В заключении представлены основные результаты работы. Авторы статьи полагают, что алгоритм может найти применение в подсистемах аутентификации информационных систем, а также в системах, в которых наиболее важным требованием является стойкость в течение длительного времени.

Ключевые слова: хеш-функция, хеш-значение, хеш, аутентификация, атака, усеченная хеш-функция, алгоритм, многоитерационное хеширование, модификатор, эквивалентно стойкая хеш-функция, итерация, криптостойкость.

Vikhman V.V., Pankov M.A. **Security increasing of hash functions in information systems on the basis of multi-iterative hashing algorithm with several modifiers**

Abstract. In this paper influence of multi-iterative hashing with several modifiers algorithm's parameters on its cryptographic persistence is considered. Relevance of multi-iterative hashing with several modifiers algorithm's application and need of research of its parameters are justified, the description of algorithm is provided. Cryptographic persistence of hash function to attacks which are not depend on algorithm is caused by its bitness, i.e. actually on the amount of unique hash values that hash function is able to generate. For an estimation of algorithm's persistence to dictionary attacks and attacks by methods of "brute force" and "birthdays" the algorithm of multi-iterative hashing with several modifiers is considered as independent hash function. Estimation of the algorithm's persistence for a given number of

iterations is offered to produce by calculating the average bitness of equivalently persistent hash function for the algorithm. The description of estimation method of algorithm's persistence is provided. The experiments are performed using a truncated cryptographically persistent hash function. The results of experiments allow to compare the algorithm's persistence metrics of under different values of its parameters. Besides, the results of the experiments allow to understand how the values of certain parameters, and combinations of values for these parameters affect for the algorithm's cryptographic persistence to dictionary attacks and attacks by methods of "brute force" and "birthdays". On the basis of the received results it is possible to draw conclusions about the values of the parameters recommended for practical application of this algorithm. In conclusion, the paper presents the main results of the work. Authors of the article believe that the algorithm can find application in authentication subsystems of information systems, and also in systems where the most important requirement is persistence for a long time.
Keywords: hash function, hash value, hash, authentication, attack, truncated hash function, algorithm, multi-iterative hashing, modifier, bitness, equivalently persistent hash function, iteration, cryptographic persistence.

Введение. В настоящее время задача защита информации является одной из наиболее важных для информационных систем. Безопасность хранения паролей в подсистемах аутентификации ИС достигается с помощью использования криптографических хеш-функций. Криптографическая хеш-функция должна удовлетворять ряду требований [1 – 5]:

– *стойкость к вычислению прообраза* – невозможность нахождения неизвестного прообраза для любых предварительно заданных хеш-значений, т.е. для заданной хеш-функции h вычислительно невозможно найти неизвестный прообраз x при предварительно заданном хеш-значении $y=h(x)$ для любого значения y (под термином «вычислительно невозможно» здесь и далее будем понимать, что алгоритм, выполняющий данное преобразование, обладает не менее чем экспоненциальной сложностью);

– *стойкость к вычислению второго прообраза* – невозможность нахождения любого другого прообраза, который давал бы такое же хеш-значение, как и заданный, т.е. для заданной хеш-функции h и прообраза x вычислительно невозможно найти другой прообраз $x' \neq x$, для которого выполнялось бы условие $h(x)=h(x')$;

– *стойкость к коллизиям* – невозможность нахождения двух прообразов, для которых вырабатывалось бы одинаковое значение, т.е. для заданной хеш-функции h вычислительно невозможно найти два прообраза x и x', $x' \neq x$, для которых выполнялось бы условие $h(x)=h(x')$.

На криптографические хеш-функции возможны следующие атаки:

– нахождение прообраза x по заданному значению $y=h(x)$;

– нахождение прообраза x' по заданному прообразу x, для которого выполняется условие $h(x)=h(x')$;

– нахождение двух прообразов x и x', $x \neq x'$, для которых выполнялось бы условие $h(x)=h(x')$.

Атака "грубой силой" [1] может быть выполнена для нахождения прообраза по заданному хеш-значению или для нахождения прообраза, дающего заданное хеш-значение. Суть атаки заключается в последовательном или случайном переборе входных сообщений и сравнения результата выполнения хеш-функции с заданным. Сложность такой атаки оценивается 2^{l-1} операций вычисления хеш-значений, где l – длина хеш-значения в битах.

Атака методом «дней рождения» [1, 6] выполняется для нахождения двух различных сообщений с одинаковыми хеш-значениями. Эта атака основана на парадоксе «дней рождения» и заключается в том, что в двух сгенерированных множествах хеш-значений, содержащих *n1* и *n2* элементов соответственно, вероятность нахождения совпадающих элементов между этими множествами оценивается следующей формулой:

$$P \approx 1 - e^{-\frac{n_1 n_2}{2^l}}. \quad (1)$$

В частности, при $n_1 = n_2 = 2^{\frac{l}{2}}$ сложность атаки оценивается как $2^{\frac{l}{2}+1}$ операций вычисления хеш-значений, а вероятность успеха равна:

$$P \approx 1 - \frac{1}{e} \approx 0,63. \quad (2)$$

Также существуют атаки, основанные на принципах атаки методом «грубой силы». Разновидностями этой атаки являются словарные атаки [7]. Название этого вида атак произошло благодаря тому, что основу множества перебираемых паролей составляют слова какого-либо языка. Словарные атаки используют присущую большинству пользователей тенденцию к использованию легко запоминаемых паролей, к которым часто относятся различные слова и их варианты (например, замена части букв слова на похожие по написанию цифры или спецсимволы). По сравнению с методом «грубой силы» словарные атаки осуществляют перебор по существенно меньшему множеству возможных значений [7, 8].

2. Постановка задачи. Для эффективного противостояния атакам методами «грубой силы» и «дней рождения» можно применять многоитерационное хеширование, которое позволяет повысить количество операций, необходимое для генерации хеш-значения, и, следовательно, усложнить указанные атаки [9]. Наиболее действенным

методом защиты от различных видов словарных атак [7, 8] является дополнение (например, путем конкатенации) хешируемого пароля какой-либо случайной величиной (модификатором), которая впоследствии хранится вместе с хешированным паролем. Следовательно, для определения пароля атакующий должен рассмотреть также все множество значений данной величины [7]. В качестве примеров функций, реализующих оба этих подхода (многоитерационное хеширование, использование модификатора), можно привести функции bcrypt и scrypt [10, 11].

Алгоритм многоитерационного хеширования с несколькими модификаторами [12], предлагаемый авторами, комбинирует в себе указанные выше методы защиты, и позволяет противостоять как атакам методами «грубой силы» и «дней рождения», так и словарным атакам. Данный алгоритм позволяет использовать несколько модификаторов, что следует из его названия. Алгоритм также может использоваться для повышения стойкости к рассмотренным атакам произвольной функции хеширования. Ввиду последнего обстоятельства актуальной задачей является анализ параметров алгоритма многоитерационного хеширования с несколькими модификаторами и их влияния на криптостойкость данного алгоритма.

3. Алгоритм многоитерационного хеширования с несколькими модификаторами. Алгоритм многоитерационного хеширования с несколькими модификаторами, предназначен для повышения стойкости применяемой в нем криптографической хеш-функции к атакам, не зависящим от алгоритма (атака «грубой силой», атака методом «дней рождения», словарные атаки) [12].

Рассмотрим алгоритм. Сначала с помощью криптостойкого генератора случайных чисел (ГСЧ) инициализируются k модификаторов разрядностью d бит каждый. Каждому модификатору присваивается индекс от *0* до *k–1* ($k \in [1; 255]$). Затем вычисляется Z-хеш [13 – 17], который будет задавать порядок выбора модификаторов:

$$Z = h\left(x + h\left(x\right)\right), \tag{3}$$

где x – входное значение, h – функция хеширования, *«+»* – операция конкатенации [12]. Хеш-значение Z представим в виде массива байтов:

$$Z = \begin{pmatrix} z_0 \\ z_1 \\ \cdots \\ z_{t-1} \end{pmatrix}. \tag{4}$$

На каждой итерации хеширования будет применяться один из модификаторов, выбираемый по индексу j:

$$j = z_{i \bmod t} \bmod k, \tag{5}$$

где i – номер итерации, t – размерность массива Z [12]. На первой итерации хеш вычисляется от конкатенации входного значения и модификатора, на последующих итерациях в качестве входного значения используется хеш, полученный на предыдущей итерации. Таким образом, входное значение будет косвенным образом определять порядок выбора модификаторов при многоитерационном хешировании. Так как инициализация модификаторов осуществляется при помощи криптостойкого ГСЧ, для повторного вычисления хеша необходимо сохранять модификаторы и порядок их следования.

Кроме количества итераций, алгоритм имеет три параметра: длина Z-хеша в битах, количество модификаторов, разрядность (битность) модификаторов. Определим, как значения этих параметров влияют на стойкость алгоритма к атакам методами «грубой силы» и «дней рождения».

Стойкость хеш-функции к таким атакам зависит от ее разрядности, т.е. фактически – от количества уникальных хеш-значений, которые способна генерировать данная хеш-функция [1, 12]. Для оценки стойкости алгоритма подадим на его вход множество, содержащее 2^b уникальных значений, где b – разрядность оцениваемой хеш-функции, а затем подсчитаем количество уникальных хешей n во множестве хеш-значений. Тогда стойкость алгоритма можно оценить через разрядность s абстрактной эквивалентно стойкой хеш-функции:

$$s = \log_2 n. \tag{6}$$

В экспериментах использовалась усеченная 16-битная хеш-функция SHA-1. Использование усеченной хеш-функции возможно ввиду того, что криптографическая хеш-функция (в данном случае это криптографическая хеш-функция SHA-1) позволяет выбирать любой набор битов результата для формирования усеченной хеш-функции с аналогичными свойствами, хотя и, естественно, меньшей стойкости – соответственно своей новой разрядности [18 – 20]. Входные данные – беззнаковые двоичные 16-разрядные числа в диапазоне от 0 до 65535. Параметры алгоритма обозначаются в формате *t/k/d*, где t – разрядность Z-хеша, k – количество модификаторов, d – разрядность модификаторов. Так как значения модификаторов являются случайными, для каждого количества итераций будем проводить серию из пяти экспериментов, а среднюю разрядность эквивалентно

стойкой хеш-функции $\bar{s}$ для текущего количества итераций будем рассчитывать следующим образом:

$$\bar{s} = \frac{1}{5}\sum_{q=1}^{5} \log_2 n_q . \qquad (7)$$

Первая серия экспериментов проводилась при использовании восьми 8-битных модификаторов при изменяемом значении разрядности Z-хеша (из 160-битного хеша SHA-1 выбиралось необходимое количество младших двоичных разрядов). Результаты представлены на рисунке 1.

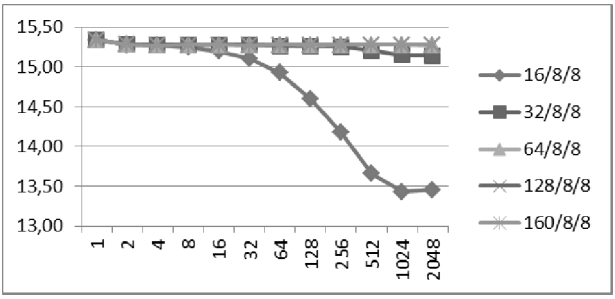

Рис. 1. Диаграммы зависимостей средних разрядностей эквивалентно стойких хеш-функций от количества итераций алгоритма (изменяемая разрядность Z-хеша, восемь 8-битных модификаторов)

Затем две серии экспериментов были проведены с использованием 16-битных и 32-битных модификаторов (рисунки 2, 3). Из полученных диаграмм видно, что с увеличением разрядности Z-хеша, при неизменном количестве и разрядности модификаторов, растут и средние разрядности эквивалентно стойких хеш-функций. Таким образом, можно сделать вывод о том, что разрядность Z-хеша напрямую влияет на стойкость алгоритма.

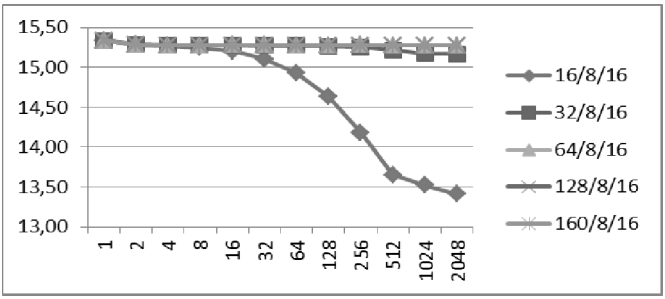

Рис. 2. Диаграммы зависимостей средних разрядностей эквивалентно стойких хеш-функций от количества итераций алгоритма (изменяемая разрядность Z-хеша, восемь 16-битных модификаторов)

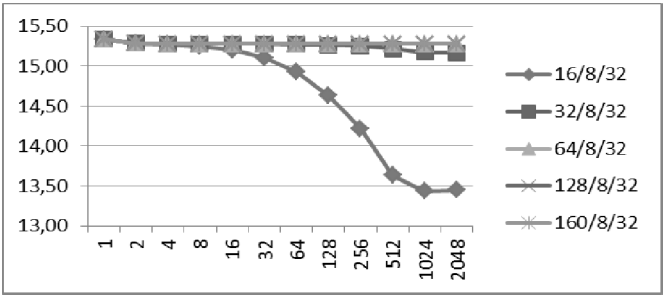

Рис. 3. Диаграммы зависимостей средних разрядностей эквивалентно стойких хеш-функций от количества итераций алгоритма (изменяемая разрядность Z-хеша, восемь 32-битных модификаторов)

Следует отметить, что разрядность модификаторов, при прочих равных параметрах алгоритма, не влияет на среднюю разрядность эквивалентно стойкой хеш-функции. При этом количество всех возможных значений набора из k модификаторов оценивается как 2^{dk}.

Также было исследовано влияние количества используемых модификаторов на стойкость алгоритма. В экспериментах использовались полные значения Z-хешей при изменяемом количестве модификаторов. Проведено три серии экспериментов с использованием 8-битных, 16-битных и 32-битных модификаторов соответственно. Полученные диаграммы представлены на рисунках 4, 5 и 6.

Из полученных результатов можно сделать вывод, что чем меньше модификаторов применяется в алгоритме, тем быстрее, с увеличением количества итераций, растет количество коллизий, и, соответственно, падает стойкость эквивалентно стойкой хеш-функции алгоритма. Для того чтобы разрядность эквивалентно стойкой хеш-функции для алгоритма равнялась разрядности используемой в нем хеш-функции, необходимо применять не менее четырех модификаторов.

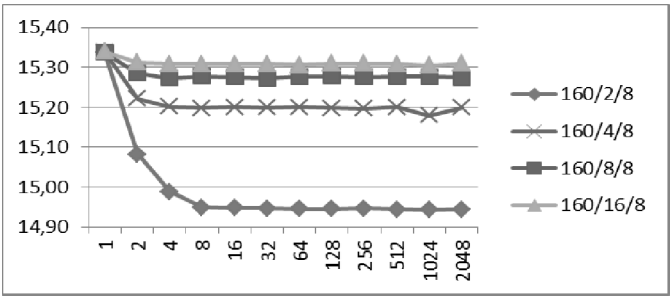

Рис. 4. Диаграммы зависимостей средних разрядностей эквивалентно стойких хеш-функций от количества итераций алгоритма (изменяемое количество 8-битных модификаторов, 160-битный Z-хеш)

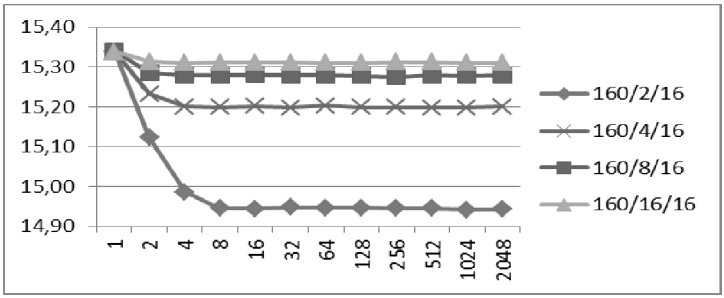

Рис. 5. Диаграммы зависимостей средних разрядностей эквивалентно стойких хеш-функций от количества итераций алгоритма (изменяемое количество 16-битных модификаторов, 160-битный Z-хеш)

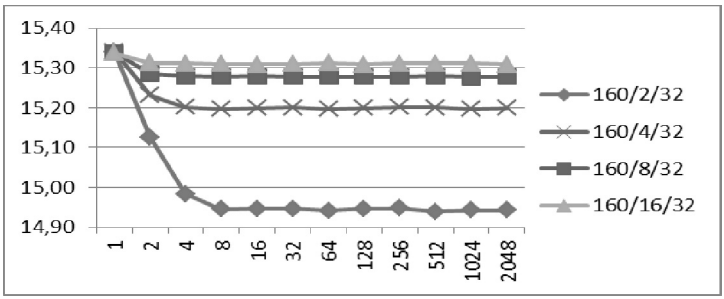

Рис. 6. Диаграммы зависимостей средних разрядностей эквивалентно стойких хеш-функций от количества итераций алгоритма (изменяемое количество 32-битных модификаторов, 160-битный Z-хеш)

Из данных диаграмм также видно, что разрядность модификаторов, при прочих равных параметрах алгоритма, не влияет на его стойкость.

4. Заключение. Алгоритм многоитерационного хеширования с несколькими модификаторами позволяет повысить трудоемкость вычисления хеш-значений с помощью существующих криптостойких хеш-функций. На стойкость алгоритма напрямую влияет разрядность Z-хеша: чем больше байтов содержит Z-хеш, тем выше средняя разрядность эквивалентно стойкой хеш-функции при одном и том же количестве итераций. Таким образом, в данном алгоритме рекомендуется использовать Z-хеш с разрядностью, равной полной разрядности применяемой в нем хеш-функции.

Количество модификаторов, применяемых в алгоритме, также влияет на его стойкость. При одинаковом количестве итераций средняя разрядность эквивалентно стойкой хеш-функции выше для алгоритма с большим количеством модификаторов. Для того чтобы не

происходило потери стойкости применяемой хеш-функции, в алгоритме необходимо использовать не менее четырех модификаторов. Разрядность модификаторов, при прочих равных параметрах алгоритма, не влияет на среднюю разрядность эквивалентно стойкой хеш-функции, но при этом влияет на сложность атак методами, не зависящими от алгоритма. Количество и разрядность модификаторов следует задавать исходя из требований к конкретной системе аутентификации.

Направление дальнейших исследований будет связано с особенностями внедрения разработанного алгоритма в подсистемы аутентификации пользователей информационных систем.

Литература

1. *Вервейко В.Н., Пушкарев А.И., Цепурит Т.В.* Функции хэширования: классификация, характеристика и сравнительный анализ. URL: http://bezopasnik.org/article/book/94.pdf.
2. *Schneier B.* One-Way Hash Functions // Dr. Dobb's journal. 1991. vol. 16. no. 9. pp. 148–151.
3. *Biham E., Shamir A.* Differential cryptoanalysis of FEAL and NHash // In Advances in Cryptology (Eurocrypt '91). 1990. pp. 1–16.
4. *Biham E.* On the Applicability of Differential Cryptoanalysis to Hash Functions // In E.I.S.S Workshop on Cryptographic Hash Functions. 1992. pp. 25–27.
5. *Quisquater J.-J., Delescaille J.-P.* How Easy is Collision Search. New results and applications to DES // In Advances in Cryptology (CRYPTO'89). 1990. vol. 435. pp. 408–415.
6. *Ohta K., Koyama K.* Meet-in-the-Middle Attack on Digital Signature Schemes // In Abstract of AUSCRYPT '90. 1990. pp. 110–121.
7. *Панасенко С.П.* Словарные атаки на хэш-функции // Мир и безопасность. 2009. № 4. С. 24–31.
8. *Коржик В.И., Пантелеева З.А.* Исследование метода радужных таблиц для восстановления паролей // Актуальные проблемы инфотелекоммуникаций в науке и образовани. II-я Международная научно-техническая и научно-методическая конференция: сб. научных статей. СПб.: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича. 2013. С. 824–829.
9. *Лёвин В.Ю.* О повышении криптостойкости однонаправленных хеш-функций / Фундаментальная и прикладная математика. 2009. Т. 15:5. С. 171–179.
10. *Provos N., Mazieres D.* A Future-Adaptable Password Scheme // The OpenBSD Project. URL: http://www.openbsd.org/papers/bcrypt-paper.pdf.
11. *Percival C., Josefsson S.* The scrypt Password-Based Key Derivation Function // IETF. 2012.
12. *Вихман В.В., Панков М.А.* Исследование криптостойкости алгоритмов многоитерационного хэширования в подсистемах аутентификации МИС // Актуальные проблемы электронного приборостроения (АПЭП–2014): тр. 12 междунар. конф.Новосибирск : Изд-во НГТУ. 2014. Т. 2. С. 193–199.
13. *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си // М.: Триумф. 2002. 610 с.
14. *Фергюсон Н., Шнайер Б.* Практическая криптография // М.: Издательский дом «Вильямс». 2005. С. 101–114.

15. *Алферов А.П., Зубов А.Ю., Кузьмин А.С. и др.* Основы криптографии: учебное пособие / 2-е изд., испр. и доп. // М.: Гелиос АРВ. 2002. 480 с.
16. *Menezes A.J., Van Oorschot C., Vanstone S.A*. Handbook of applied cryptography // CRC Press. Boca Raton. New York. 1997.
17. Введение в криптографию / Под общ. ред. В. В. Ященко. 4-е изд., доп. // М.: МЦНМО. 2012. 348 с.
18. *Rivest R.* The MD5 Message Digest Algorithm // RFC 1321. 1992.
19. National Institute of Standards and Technology. Secure Hash Standard // FIPS PUB 180-1. 1995.
20. National Institute of Standards and Technology. Secure Hash Standard (draft) // DRAFT FIPS PUB 180-2. 2001.

References

1. Vervejko V.N., Pushkarev A.I., Cepurit T.V. *Funktsii kheshirovaniia: klassifikatsiia, kharakteristika i sravnitel'nyi analiz* [Hashing functions: classification, characterization and comparative analysis] Available at http://bezopasnik.org/article/book/94.pdf (accessed 12.12.2014) (In Russ.).
2. Schneier B. One-Way Hash Functions. Dr. Dobb's journal. 1991. vol. 16. no. 9. pp. 148–151.
3. Biham E., Shamir A. Differential cryptoanalysis of FEAL and NHash. In Advances in Cryptology (Eurocrypt ’91). 1990. pp. 1–16.
4. Biham E. On the Applicability of Differential Cryptoanalysis to Hash Functions. In E.I.S.S Workshop on Cryptographic Hash Functions. 1992. pp. 25–27.
5. How Easy is Collision Search. New results and applications to DES. In Advances in Cryptology (CRYPTO’89). 1990. vol. 435. pp. 408–415.
6. Ohta K., Koyama K. Meet-in-the-Middle Attack on Digital Signature Schemes. In Abstract of AUSCRYPT ’90. 1990. pp.110–121.
7. Panasenko S. [Dictionary attacks on hash functions]. *Mir i bezopasnost' – World and security*. 2009. no. 4, pp. 24–31. (In Russ.).
8. Korzhik V., Panteeleeva Z. [Rainbow Tables Method for Cryptanalytic Time Memory Trade-Offs]. *Aktual'nye problemy infotelekommunikacij v nauke i obrazovani. II-ja Mezhdunarodnaja nauchno-tehnicheskaja i nauchno-metodicheskaja konferencija: sb. nauchnyh statej* [Actual problems of intelecommunications in science and education. II international scientific-technical and scientific-methodical conference: collection of scientific articles]. St. Petersburg. 2013. pp. 824–829. (In Russ.).
9. Ljovin V. [About increasing of cryptographic security unidirectional hash functions]. *Fundamental'naja i prikladnaja matematika – Fundamental and applied mathematics*. 2009. vol. 15:5. pp. 171–179. (In Russ.).
10. Provos N., Mazieres D. A Future-Adaptable Password Scheme. Available at http://www.openbsd.org/papers/bcrypt-paper.pdf (accessed 12.12.2014)
11. Percival C., Josefsson S. The scrypt Password-Based Key Derivation Function. IETF. 2012.
12. Vikhman V., Pankov M. [Research of multi-iterative hashing algorithms' cryptographic persistence in authentication subsystems of MIS]. *Trudy 12 mezhdunarodnoi konferentsii «Aktual'nye problemy elektronnogo priborostroeniia (APEP–2014)»* [Proc. 12th Int. Conf. “Actual problems of electronic instrument engineering (APEIE–2014)”]. Novosibirsk. 2014. pp. 193–199. (In Russ.).
13. Schneier B. *Prikladnaja kriptografija. Protokoly, algoritmy, ishodnye teksty na jazyke Si* [Applied Cryptography. Protocols, algorithms, and Source Code in C]. Wiley. 1995. 662 p. (Russ. ed.: Shnaier B. Prikladnaia kriptografiia. Protokoly, algoritmy, iskhodnye teksty na iazyke Si. Moskow. Triumf Publ. 2002. 610 p.).

14. Ferguson N., Schneier B. *Prakticheskaja kriptografija* [Practical Cryptography]. Wiley. 2003. 432 p. (Russ. ed.: Fergiuson N., Shnaier B. Prakticheskaia kriptografiia. Moskow. Izdatel'skii dom «Vil'iams» Publ. 2005. 424 p.).
15. Alferov A., Zubov A., Kuz'min A., Cheremushkin A. *Osnovy kriptografii* [Foundations of cryptography]. Moscow. Gelios ARV Publ. 2002. 480 p. (In Russ.).
16. Menezes A.J., Van Oorschot P.C., Vanstone S.A. Handbook of applied cryptography. CRC Press. Boca Raton. New York. 1997. 816 p.
17. Iashchenko V. *Vvedenie v kriptografiiu* [Introduction to cryptography]. Moscow. MTsNMO. 2012. 348 p. (In Russ.).
18. Rivest R. The MD5 Message Digest Algorithm. RFC 1321. 1992.
19. National Institute of Standards and Technology. Secure Hash Standard. FIPS PUB 180-1. 1995.
20. National Institute of Standards and Technology. Secure Hash Standard (draft). DRAFT FIPS PUB 180-2. 2001.

Вихман Виктория Викторовна — к-т техн. наук, доцент, заведующий кафедрой интеграционных информационных систем Новосибирского государственного технического университета (НГТУ), доцент кафедры вычислительной техники факультета автоматики и вычислительной техники Новосибирского государственного технического университета (НГТУ). Область научных интересов: автоматизированные системы управления, системы информационной безопасности. Число научных публикаций — 58. vvv@vt.cs.nstu.ru, www.nstu.ru; НГТУ, 630073, г. Новосибирск, пр. Карла Маркса, 20, РФ; р.т. +7(383)3460492, +7(383)346-0219.

Vikhman Victoria Victorovna — Ph.D, associate professor, Head of Department of information systems' integration, Novosibirsk State Technical University (NSTU), associate professor, Department of Computer Engineering, Faculty of Automation and Computer Engineering, Novosibirsk State Technical University (NSTU). Research interests: automated control systems, information security systems. The number of publications — 58. vvv@vt.cs.nstu.ru, www.nstu.ru; NSTU, 20, prospect Karla Marksa, Novosibirsk, 630073, Russian Federation; office phones +7(383)346-0492, +7(383)346-0219.

Панков Максим Александрович — аспирант кафедры вычислительной техники факультета автоматики и вычислительной техники Новосибирского государственного технического университета. Область научных интересов: системы информационной безопасности, алгоритмы и модели управления персоналом в интегрированной информационной среде. Число научных публикаций — 4. hm.mobile@mail.ru; НГТУ, 630073, г. Новосибирск, пр. Карла Маркса, 20, РФ; р.т. +7(383)3460492, +7(383)346-0219.

Pankov Maksim Aleksandrovich — Ph.D. student, Department of Computer Engineering, Faculty of Automation and Computer Engineering, Novosibirsk State Technical University (NSTU). Research interests: information security systems, algorithms and models of human resource management in integrated information environment. The number of publications — 4. hm.mobile@mail.ru; NSTU, 20, prospect Karla Marksa, Novosibirsk, 630073, Russian Federation; office phones +7(383)346-0492, +7(383)346-0219.

РЕФЕРАТ

Вихман В.В., Панков М.А. **Повышение стойкости хеш-функций в информационных системах на основе алгоритма многоитерационного хеширования с несколькими модификаторами.**

В статье рассматривается анализ параметров алгоритма многоитерационного хеширования с несколькими модификаторами, разработанного авторами. Главной особенностью алгоритма является возможность его применения для повышения стойкости различных криптографических хеш-функций к атакам, не зависящим от алгоритма. Следующие параметры задают настройки алгоритма: количество итераций, количество байт Z-хеша, количество модификаторов, разрядность (битность) модификаторов. Эксперименты производятся для оценки влияния значений различных параметров алгоритма на стойкость применяемой в нем хеш-функции. Оценка производится путем вычисления средних разрядностей эквивалентно стойких хеш-функций. Из результатов экспериментов следует, что количество байт Z-хеша напрямую влияет на стойкость хеш-функции. Разрядность Z-хеша следует выбирать равной полной разрядности применяемой хеш-функции. Также влияние на стойкость хеш-функции оказывает количество применяемых в алгоритме модификаторов. Увеличение количества модификаторов, при прочих равных значениях параметров алгоритма, приводит к повышению стойкости хеш-функции. Разрядность модификаторов на среднюю разрядность эквивалентно стойкой хеш-функции не влияет. Параметры алгоритма необходимо подбирать, исходя из требований к конкретной системе аутентификации.

SUMMARY

Vikhman V.V., Pankov M.A. **Security increasing of hash functions in information systems on the basis of multi-iterative hashing algorithm with several modifiers.**

The article is considers the analysis of multi-iterative hashing with several modifiers algorithm's parameters, developed by the authors. The main feature of the algorithm is the possibility of its application to increase the security of different cryptographic hash functions to attacks that do not depend on the algorithm. The following parameters specify the settings of the algorithm: the number of iterations, the number of bytes in the Z-hash, the number of modifiers, the bitness of modifiers. The experiments are carried out to estimate the impact of different values for the algorithm's parameters on the security of hash function applied in it. The estimation is carried out by calculating the average bitnesses of equivalently persistent hash functions. From the experimental results it follows that the number of bytes in the Z-hash directly affects for security of the hash function. Bitness of Z-hash should be chosen equal to the full bitness of the used hash function. Also the impact to security of the hash function provides by the number of modifiers used in the algorithm. The increasing in the number of modifiers, under other equal values of algorithm's parameters, increases the security of the hash function. The bitness of the modifiers on the average bitness of equivalently persistent hash function has no effect. The parameters of algorithm should be selected based on requirements to specific authentication system.