УДК 681.3.06:519.248.681

**В.Е. ЧЕВАРДИН, И.Н. ПОНОМАРЕВ, В.Г. ПРОКОПЕНКО**

***Военный институт телекоммуникаций и информатизации Национального технического университета Украины «КПИ», Полтава, Украина***

# ОЦЕНКА СТОЙКОСТИ MAC-КОДОВ И ПЕРСПЕКТИВНЫЕ ПУТИ РАЗВИТИЯ

*Рассмотрены теоретические аспекты построения и применения MAC-кодов в телекоммуникационных системах на основе теоретико-сложностных задач математики. Проведен анализ современных хеш-функций, таких как HMAC, UMAC и т.д., основным недостатком которых является невозможность получения доказуемо-стойкой схемы аутентификации сообщений. Предложен подход к построению теоретически стойких MAC-алгоритмов для выполнения различных криптографических задач без роста вычислительной сложности криптопреобразований.*

***Ключевые слова:*** *целостность, аутентификация, хеш-функция, ключ, криптостойкость.*

## Введение

Известно, что неотъемлемой составляющей любой современной телекоммуникационной системы является подсистема криптографической защиты информации. Разработка и эксплуатация криптосистемы является важнейшей задачей при проектировании современной телекоммуникационной системы [1]. В связи с чем, эффективность ее решения зависит от криптографической стойкости криптопреобразований системы и их вычислительной сложности. Известно, что количество зашифрованных данных, доступных криптоаналитику, напрямую влияют на эффективность дифференциального и частотного криптоанализа. В следствии чего, возникает одно из противоречий, когда в условиях стремительного возрастания объемов шифрованой информации, передаваемой по сетям, используемые на практике криптосистемы не позволяют обеспечивать требуемую стойкость к современным методам криптоанализа [1, 2].

Одним из примеров криптосистем, которые не выдерживают современных тенденций развития информационных технологий и возможностей криптоаналитиков являются механизмы аутентификации сообщений с использованием ключевого хеширования – Message Authentication Code (MAC-кодов) [2].

Результаты разработок в этой области за последние 10 лет представляют немалый арсенал методов и алгоритмов ключевого хеширования [2]. Согласно теории аутентификации [3, 4] их можно разделить на три уровня:

1. Вычислительная стойкость;
2. Доказуемая стойкость;
3. Безусловная стойкость.

Как показывает практика использование криптосистем с ключами, длина которых не меньше длины передаваемого сообщения (одно из требований безусловной стойкости) не оправдано. На практике, как правило, используют вычислительно стойкие схемы ключевого хеширования (MD5 (RFC 1321), SHA-1 (FIPS PUB 180-1), SHA-256 и SHA-512) и реже схемы с доказуемой стойкостью (MASH-1, MASH-2, BLS). Рассмотрим более детально структуру алгоритма формирование MAC-кода и требования к ним.

## 1. Алгоритм формирования MAC-кодов, показатели и характеристики

Хеш-код (или образ сообщения) представляет собой короткую цифровую строку (по современным требованиям 128 или 160 битов), формируемую из сообщения произвольной длины по специальному алгоритму [2].

Алгоритмы бывают однонаправленные – *бесключевые* (MD5, SHA, ГОСТ 28147-89, ГОСТ 34.311-95.) – они обеспечивают лишь целостность и зависят от свойств блочно-симметричного алгоритма, лежащего в их основе. Для таких схем основным показателям является вероятность коллизии, когда хеш-код двух отличающихся сообщений совпадает:

$$h_k(M_1) = h_k(M_2), M_1 \neq M_2, \quad (1)$$

где $h_k$ – k-тое правило хеширования,

$M_1$, $M_2$ – открытые сообщения.

Вероятность коллизии в лучшем случае равна:

$$P_{coll} = 2^{-l_h}, \quad (2)$$

где $l_h$ – длина хеш-кода.

К примеру, алгоритм MD5 для входных сообщений произвольной длины генерирует на выходе

128-битовый. SHA-1 – 160-битовый хеш-код. Следовательно, $P^{MD5}_{coll} = 2^{-128}$, $P^{SHA}_{coll} = 2^{-160}$ в лучшем случае.

Определение коллизии в схеме SHA-1 привело к появлению усовершенствованных алгоритмов: SHA-256 и SHA-512, генерирующих строки в 256 и 512 бит. Однако сегодня уже найдена коллизия и для 22 раундов SHA-256, что дает возможность в ближайшем будущем получить коллизию для всех раундов как SHA-256, так и для SHA-512 [1].

Оценка статистических свойств этого семейства хеш-функций дает возможность прогнозировать вероятность появления алгоритма определения коллизий. Для оценки используется статистический тест, разработанный в NIST (Special Publication 800-22) – Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. Это подчеркивает неоднозначность обоснования стойкости хеш-алгоритма. Такие схемы аутентификации позволяют обеспечить лишь вычислительную стойкость аутентификации.

*Ключевые* хеш-функции (MAC-алгоритмы) обеспечивают как целостность, так и аутентичность электронных сообщений. Это обеспечивается секретностью ключа аутентификации. Стойкость MAC-алгоритмов зависит, во-первых, от ключевой стойкости, во-вторых, от вероятности коллизии, которая косвенно зависит и от параметров ключа. В целом основным показателем стойкости аутентификации принято считать вероятность обмана. Гилбертом, Мак-Уильямсом и Слоуном [3, 4] была получена нижняя граница вероятности успешного обмана:

$$P_d \geq \frac{1}{\sqrt{|\varepsilon|}}, \qquad (3)$$

где $|\varepsilon|$ – общее число правил кодирования (число ключей хеширования).

При наличии этой границы для оценки MAC-алгоритмов возникают вопросы: как обеспечить вероятность успешного обмана не выше данной границы? Либо как обеспечить наименьшую верхнюю границу вероятности успешного обмана? Для ответов на эти вопросы в работах [3, 4] было предложено представлять хеш-функцию в виде универсального класса функций ε - U(N, n, r), где N – количество функций отображения множества открытых текстов $\sum^n$ мощности n в множество хеш-кодов $\sum^r$ мощности r. Причем для двух различных элементов $(M_1, M_2) \in \sum^n$ существует не больше, чем $N \cdot \varepsilon$ функций $f \in H$ таких, что $f(M_1) = f(M_2)$. Примерами являются классы хеш-функций, построенные на основе полиномиальных функций, на основе ортогональных массивов, детальное исследование которых представлено в [5]. Применение данного подхода позволило теоретически обосновать верхние границы вероятности коллизий и получить целое семейство хеширующих функций, в частности и MAC-алгоритмов.

На конкурсах криптографических систем, проведенных за последние 8 лет, среди MAC-алгоритмов лидирующее место заняли функции HMAC, UMAC, TTMAC, RMAC, EMAC [2]. Общая схема MAC-алгоритма представлена на рис.1. Основная идея в построении всего семейства вычислительно стойких MAC-алгоритмов состоит в использовании вычислительно стойкого генератора сеансовых ключей $k_i = f(K)$. Примерами таких функций являются алгоритмы: DES, TDES, AES, RIPEMD, SHA-1, SHA2, MD5.

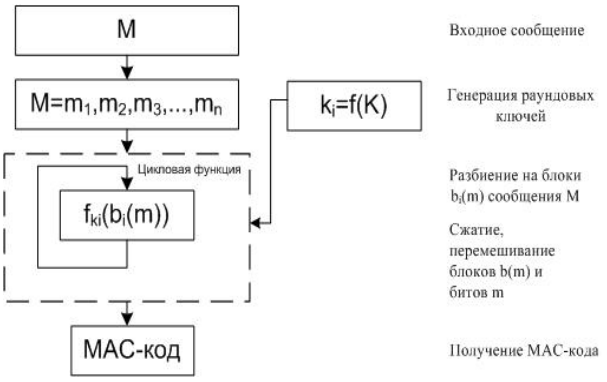


Рис. 1. Общая схема MAC-алгоритма

Согласно подхода, используемого в данных алгоритмах, вычислительная стойкость MAC-алгоритма сводится к стойкости используемого генератора раундовых ключей. Убедимся в этом.

Предположим, что $l_k > l_{MAC}$, т.е. длина ключа больше длины MAC. Тогда, зная сообщение M1 и его $MAC_1 = h_k(M_1)$, криптоаналитик может вычислить $MAC_i = h_{k_i}(M_1)$ для всех возможных ключей ki. При этом, по крайней мере, для одного из ключей kr будет получено совпадение MACr = MAC1. Криптоаналитик вычислит 2k значений MAC, тогда как при длине l MAC битов существует всего $2^{l_{MAC}}$ MAC-кодов. Учитывая, что $l_k > l_{MAC}$ ситуация $MAC_r = MAC_1$ получится для нескольких kr. В среднем совпадет для 2k/2n = 2(k-n) ключей. Поэтому для вычисления единственного ключа криптоаналитику требуется знать несколько пар сообщение и соответствующий ему MAC.

Таким образом, простой перебор всех ключей требует не меньше, а больше усилий, чем поиск ключа симметричного шифрования той же длины.

### 1.1. Схема UMAC

Известным вариантом построения MAC-алгоритма является схема на рис. 2. Например, для выработки ключа в алгоритме UMAC используется распространенный шифр AES. Схема EMAC также реализовывается с использованием AES.

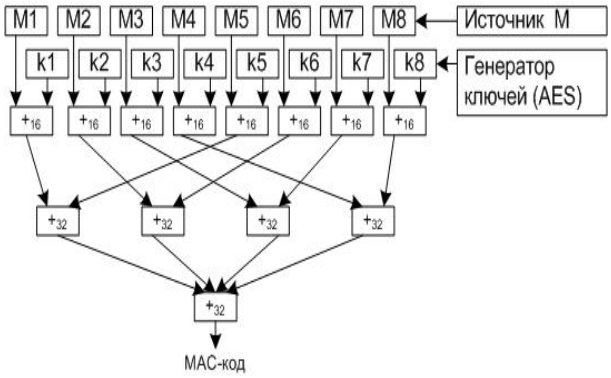


Рис. 2. Схема UMAC32

Следовательно, ключевая стойкость UMAC сводится к стойкости шифра AES. Математическая запись функции UMAC32 представлена выражением:

$$UMAC(K,M,Nonce) = (UHASH(K,M) + \\ + PDF(K,Nonce)) \bmod 2 \quad (4)$$

Это в свою очередь подтверждает, что MAC-алгоритм обладает вычислительной стойкостью и как следствие не позволяет доказать теоретически вероятность коллизии.

В существующих реализациях MAC-алгоритмов CBC MAC Bellare, Kilian, и Rogaway предложили EMAC.

$$EMAC_{K1,K2}(M) = E_{K2}(CBC_{K1}(M)), \quad (5)$$

где M – это сообщение,

K1 - это ключ CBC MAC;

$CBC_{K1}(M)$ – это значение CBC MAC M.

Однако, EMAC требует двух ключевых последовательностей лежащих в основе блочного шифра.

### 1.2. Схема XCBC

Следующий Black и Rogaway предложили XCBC, который требует только одной ключевой последовательности лежащей в основе блочного шифра. XCBC берет три ключа: один ключ блочного шифра, и два ключа n-bit K2 и K3. XCBC описан на рис. 1.

### 1.3. Схема TMAC

Окончательно Kurosawa и Iwata предложили двухключевой CBC MAC (TMAC).

TMAC получен от XCBC с помощью замены (K2, K3) на (K2*u, K2), где u – это некоторая ненулевая константа и "*" означает умножение в $GF(2^n)$.

### 1.4. Схема OMAC

OMAC (рис.3) берет только один ключ, K блочного шифра. OMAC - это общее имя для OMAC1 и OMAC2. OMAC1 получен от XCBC с помощью замены (K2, K3) на $(L*u, L*u^2)$ для некоторой ненулевой постоянной u в $GF(2^n)$, где L представляется $L = EK(0^n)$. OMAC2 так же получается с помощью использования $(L*u, L*u^{-1})$.

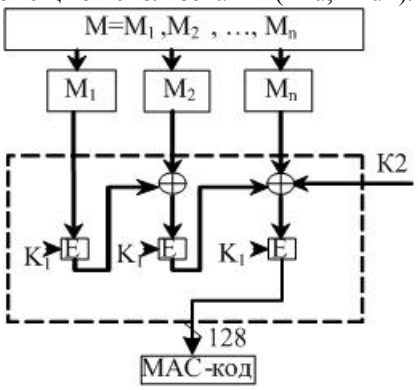


Рис. 3. Способ построения MAC-алгоритма на основе XCBC

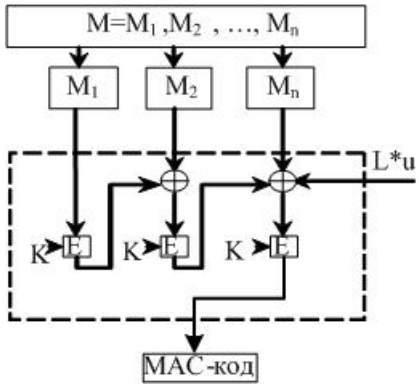


Рис. 3. Иллюстрация OMAC1

В TMAC, K2 - это часть ключа, пока в OMAC, L не является частью ключа и генерируется от K.

(PMAC – это OMAC в режиме OCB для $L = E_K(0^n)$ также используется в качестве ключа универсальная функция хеширования. Однако, L появляется, как выход некоторого внутреннего блочного шифра только с незначительной вероятностью.)

### 1.5. Схема RMAC

aulmes, Joux и Valette предложили RMAC, который является расширением EMAC.

RMAC кодирует значение CBC MAC с $K2 \oplus R$, то есть:

$$RMAC_{K1,K2}(M) = (E_{K2} \oplus R(CBC_{K1}(M)), R) \quad (6)$$

где R – это n-битная случайная строка – часть тега [2].

Таким образом, повышение стойкости MAC-алгоритма сводится к теоретическому обоснованию стойкости ключевого генератора, т.е. доказательству его криптографической стойкости.

Одним из вариантов обеспечения доказуемой стойкости MAC-алгоритма является использование стойкого генератора раундовых ключей. Один из вариантов представлен на рис. 4.

Для этого в качестве генератора ключевых последовательностей необходимо использовать функцию $f_k(Q)$, которая основывается на одной из теоретико-сложностных задач:

1. Криптопреобразования в простом поле;
2. Криптопреобразования в группе точек EC;
3. Криптопреобразования в группе дивизоров точек EC.

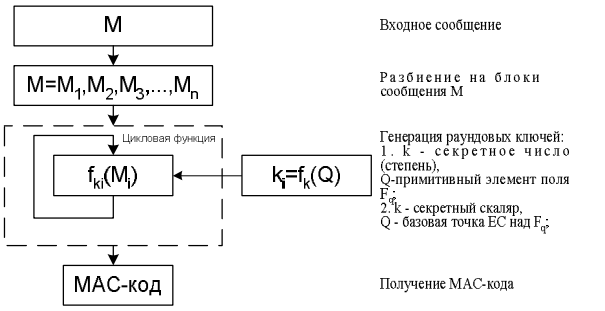


Рис. 4. Схема перспективного MAC-алгоритма

Эти криптопреобразования позволяют обеспечить доказуемую стойкость MAC-алгоритма.

## 2. Способ усовершенствования алгоритма HMAC

В настоящий момент одним из широко используемых методов формирования MAC-кода является HMAC (Keyed Hashing for Message Authentication Code. HMAC устраняет такие недостатки алгоритма MD5 как случайная коллизия.

В алгоритме HMAC хеш-функция представляет собой "черный ящик". Это, во-первых, позволяет использовать существующие реализации хеш-функций, а во-вторых, обеспечивает легкую модернизацию существующей хеш-функции. Проведем анализ реализации алгоритма HMAC с целью оценки его стойкости.

Введем следующие обозначения:

- H – встроенная хеш-функция;
- b – длина блока используемой хеш-функции;
- n – длина хеш-кода;
- K – секретный ключ, полученный с помощью криптопреобразований в группе точек EC. К ключу слева добавляют нули, чтобы получить b-битовый ключ $K^+$.

Вводится два вспомогательных значения:

- Ipad – значение 00110110, повторенное b/8 раз;
- Opad – значение 01011010, повторенное b/8 раз.

Функцию HMAC можно представить выражением (7):

$$HMAC = H((K^+ \oplus Opad) \| H((K^+ \oplus Ipad) \| M)) \quad (7)$$

Алгоритм вычисления представлен на рис. 5.

Согласно алгоритму необходимо выполнить следующие шаги:

1. Значение K слева дополняется нулями, чтобы получить b-битовую строку $K^+$ (например, если K имеет длину 160 бит и b = 512, то значение K будет дополнено 44 нулевыми байтами 0x00).
2. Значение $K^+$ складывается операцией XOR с Ipad, в результате чего получается b-битовый блок $S_i$.
3. $S_i$ дополняется блоком M.
4. К потоку, полученному в 3, применяется функция H.
5. Значение $K^+$ связывается операцией XOR с Opad, в результате чего получается b-битовый блок $S_o$.
6. Результат хеширования, полученный в 4, дополняется к $S_o$.
7. К потоку, полученному в 6, применяется функция H, и результат подается на выход.

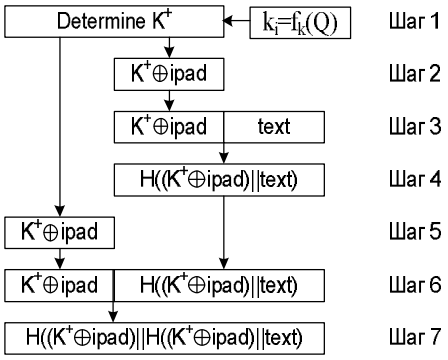


Рис. 5. Структура MAC-алгоритма

Связывание с ipad означает переключение половины битов K. Точно так же связывание с opad означает переключение половины битов K, но для другого набора битов. Поэтому в результате применения к $S_i$ и $S_o$ алгоритма хеширования из K получается два ключа, сгенерированных псевдослучайным образом.

Для достаточно длинных сообщений алгоритм HMAC должен выполняться приблизительно за время работы встроенной функции хеширования. В HMAC дополнительно требуется применить базовую функцию хеширования три раза (для $S_i$, $S_o$ и блока, получаемого при внутреннем хешировании).

Для аутентификации с использованием HMAC, к сообщению прилагается дайджест, сгенерированный с помощью HMAC и секретного ключа K, известного обоим контрагентам. Получатель может сгенерировать дайджест пришедшего от отправителя сообщения и по идентичности с приложенным к

сообщению дайджестом убедиться, что получил сообщение именно от отправителя, и это сообщение не было изменено в процессе передачи.

## Заключение

Таким образом, полученные в работе результаты отражают новый подход к построению ключевых схем хеширования – MAC-алгоритмов. Основой для создания новых схем послужили теоретико-сложностные задачи математики. Это дает возможность теоретически обосновать границу вероятности коллизий и получить доказуемо-стойкую схему аутентификации сообщений, что не позволяют существующие алгоритмы: HMAC, UMAC, TTMAC, RMAC, EMAC.

В отличии от известных схем новые схемы хеширования могут незначительно повысить вычислительную сложность криптопреобразований, однако это не превзойдет по сложности современные методы электронной цифровой подписи. Данный недостаток может быть компенсирован созданием единой библиотеки криптографических функций: для электронной цифровой подписи, поточного шифрования, генерации общих секретных ключей и обеспечения аутентификации с использованием MAC-алгоритмов.

## Литература

*1. Смарт Н. Криптография / Н. Смарт. - Москва: Техносфера, 2005. – 528 с.*

*2. Final report of European project namber IST-1999-12324, named New European Schemes for Signatures, Integrity and Encryption. - April 19, 2004 – Version 0,15 (beta).*

*3. Ohta K. Meet-in-the-Middle Attack on Digital Signature Schemes / K. Ohta, K. Koyama // In Abstract of AUSCRYPT '90. - 1990.*

*4. Bellare M. Keying hash function for message authentication. / M. Bellare, R. Canetti, H, Krawczyk. – 1996. – P. 1-15.*

*5. Black J. Rogaway UMAC: Fast and provably secure message authentication / J. Black, S. Halevi, H. Krawczyk. - Springer-Verlag, 1999. – P. 216-233.*

*Поступила в редакцию 15.02.2009*

**Рецензент:** д-р техн. наук, проф., А.А. Смердов, Полтавская государственная аграрная академия, Полтава, Украина.

### ОЦІНКА СТІЙКОСТІ MAC-КОДІВ ТА ПЕРСПЕКТИВНІ ШЛЯХИ РОЗВИТКУ

***В.Є. Чевардін, І.М. Пономарев, В.Г. Прокопенко***

Розглядаються теоретичні аспекти побудови та застосування MAC-кодів. Проведений аналіз сучасних хеш-функцій, на принципі дії яких засновані представлені MAC-коди. Запропонований спосіб побудови HMAC-коду, котрий дозволяє забезпечити цілісність повідомлень, встановлення їх автентичності та запобігання повторного використання, а також напрямок подальшої модернізації даного виду кодів і галузі їх застосування

**Ключові слова:** цілісність, автентифікація, хеш-функція, ключ, криптостійкість.

### ESTIMATION OF FIRMNESS OF MAC-CODES AND PERSPECTIVE PATHS OF DEVELOPMENT

***V.E. Chevardin, I.N. Ponomarev, V.G. Prokopenko***

The theoretical aspects of application of MAC-codes are examined. The analysis of modern hash-functions is conducted, on principle of action of which the examined MAC-codas are based. The method of construction of HMAC-code, which allows to provide integrity of reports, establishment of their authenticity and prevention of the repeated use, is resulted, and also direction of further modernization of the given type of codes and sphere of their application is specified.

**Keywords:** integrity, authenticity, hash-function, key, cryptofirmness.

**Чевардин Владислав Евгеньевич** – канд. техн. наук, начальник кафедры военных телекоммуникационных сетей и защиты информации Военного института телекоммуникаций и информатизации национального технического университета Украины «КПИ», Полтава, Украина, e-mail: chevardin_vlad@mail.ru.

**Пономарев Игорь Николаевич –** заместитель начальника факультета Военных средств связи по научной и учебной работе Военного института телекоммуникаций и информатизации национального технического университета Украины «КПИ», Полтава, Украина.

**Прокопенко Вадим Григорьевич –** преподаватель кафедры беспроводных технологий в военных телекоммуникационных системах и сетях Военного института телекоммуникаций и информатизации национального технического университета Украины «КПИ», Полтава, Украина, e-mail: prokvad@rambler.ru.