КИБЕРНЕТИКА И МЕХАТРОНИКА

УДК 62-192

ЗАЩИТА ДАННЫХ С ИСПОЛЬЗОВАНИЕМ КОДОВ АУТЕНТИФИКАЦИИ СООБЩЕНИЙ

Т.А. Билык[1], А.А. Внуков[2]

[1]Кафедра информационной безопасности
Московский государственный институт электроники и математики
Б. Трехсвятительский пер., 3/12, стр. 8, Москва, Россия 109028

[2]Кафедра кибернетики и мехатроники
Инженерный факультет
Российский университет дружбы народов
ул. Орджоникидзе, 3, Москва, Россия, 115419

В статье предлагаются три алгоритма кода аутентификации сообщений, обеспечивающие подлинность и целостность передаваемых данных в системах доверяющих друг другу абонентов. Приводится описание программной реализации алгоритмов и сравнительный анализ их эффективности.

Ключевые слова: код аутентификации сообщений, подлинность, целостность.

1. АКТУАЛЬНОСТЬ ТЕМЫ

Коды аутентификации сообщений используются как средство, гарантирующее подлинность источника данных и целостность самих данных при их передаче и хранении в системах доверяющих друг другу пользователей. Коды аутентификации сообщений могут применяться в информационных банковских системах, например ЦФТ-Банк, системах автоматизированного управления подвижными объектами и т.п. не только для обнаружения случайных ошибок в наборах данных, возникающих при их хранении и передаче, как при использовании хеш-функции, но и сигнализации об активных атаках злоумышленника, пытающегося осуществить навязывание ложной информации.

В подобных системах для каждого набора данных вычисляется значение кода аутентификации: на вход поступают два аргумента (сообщение и известный отправителю и получателю секретный ключ), на выходе выдается результат фиксированной длины, называемый имитовставкой, который передается или хранится

вместе с данными. При получении данных пользователь вычисляет значение имитовставки и сравнивает ее с имеющимся контрольным значением. Несовпадение говорит о том, что данные были изменены либо подделаны.

За рубежом эта тема изучалась с 1970-х гг. и сейчас быстро развивается. Предложен ряд подходов построения кодов аутентификации, несколько из них стандартизированы NIST (National Institute of Standards). Иная ситуация сложилась в нашей стране. Алгоритм ГОСТ 28147-89 в режиме выработки имитовставки [8], являющийся алгоритмом кода аутентификации, сегодня не удовлетворяет требованиям безопасности, так как длина формируемых имитовставок равна 32 битам и злоумышленник за короткий период времени может реализовать атаку полного перебора. Помимо данного устаревшего алгоритма в России нет стандартов кодов аутентификации, а также нет серьезных исследований по этой теме, за исключением работы [9]. В связи с этим стоит проблема разработки российского алгоритма кода аутентификации построения имитовставки достаточной длины для обеспечения заданного уровня безопасности. Если с точки зрения стойкости алгоритма чем длиннее используются ключ и имитовставка, тем лучше, то с точки зрения стоимости эксплуатации системы все наоборот: ограничение на верхнее значение длины ключа и имитовставки накладывают такие показатели, как пропускная способность канала, объем доступной памяти и т.п. Необходимо учитывать срок актуальности защищаемой информации. Оптимальные значения этих параметров могут быть различны для различных систем.

2. ПОСТАНОВКА ЗАДАЧИ

Необходимо разработать гибкий алгоритм кода аутентификации сообщений, позволяющий строить имитовставки различной длины, что позволило бы применять его в различных системах. Работа всех алгоритмов должна быть протестирована экспериментально.

3. ТРЕБОВАНИЯ К КОДАМ АУТЕНТИФИКАЦИИ СООБЩЕНИЙ

Детальное описание атак на коды аутентификации сообщений приведено в [7]. Для кода аутентификации сообщений с длиной ключа t битов и имитовставкой длины n битов сложность реализации атаки полного перебора на ключ оценивается как $O(2^t)$, а сложность реализации атаки полного перебора на имитовставку с учетом парадокса задачи о днях рождения как $O(2^{n/2})$. Таким образом, сложность реализации атаки на код аутентификации сообщений можно оценить как $O(2^{\min(n/2,t)})$. Исходя из этого, для обеспечения m-битного уровня безопасности половина длины имитовставки должна быть не менее $2m$, а длина ключа не менее m. До недавнего времени 128-битный уровень безопасности (длина ключа 128 битов, длина имитовставки 256 битов) считался достаточным, однако в связи с ростом вычислительных способностей предвидится увеличение этих параметров. Поэтому разрабатываемый алгоритм должен позволять вычислять имитовставки длины больше или равной 256 битам на ключе не менее 128 битов. Более подробно требования к кодам аутентификации описаны в [5].

4. РАЗРАБОТКА АЛГОРИТМА КОДА АУТЕНТИФИКАЦИИ СООБЩЕНИЙ

Одним из самых распространенных зарубежных подходов построения кодов аутентификации является построение кода аутентификации на основе алгоритма блочного шифрования. Зарубежный стандарт, реализующий данный подход, называется CMAC (Cipher Message Autentication Code) и описан в документе SP 800-38B [11].

Другой распространенный подход построения кодов аутентификации основан на однонаправленной хеш-функции, возможность его применения рассмотрена в [6].

В алгоритме используется блочный шифр в режиме сцепления блоков шифротекста. При этом используется секретный ключ, известный отправителю и получателю, и два рабочих ключа, вычисляемых от него. Перед шифрованием последнего блока он складывается операцией XOR с одним из рабочих ключей в зависимости от того, является ли последний блок текста полным либо дополняется нулевыми байтами. Последний шифрованный блок берется в качестве имитовставки.

В стандарте SP 800-38B в качестве алгоритма шифрования предлагается использовать алгоритм Advanced Encryption Standard (AES), хотя также может использоваться любой другой алгоритм блочного шифрования, например алгоритм блочного шифрования ГОСТ 28147-89. Однако при такой реализации длина имитовставки будет равна длине выходного блока шифра, т.е. 64 битам, что не удовлетворяет требованиям, приведенным в п. 3. Поэтому было принято решение создать алгоритм блочного шифрования, основанный на ГОСТ 28147—89 и имеющий длину блока достаточную для обеспечения заданного уровня безопасности (обязательна возможность обеспечения 128-битного уровня безопасности и выше). Для этого алгоритм блочного шифрования, который предлагается использовать, оперирует с блоками любой длины кратной 64 битам.

4.1. Основной алгоритм вычисления кода аутентификации сообщений

Детальное описание алгоритма приводится в [4]. Данный алгоритм на вход получает секретный ключ K длины 256 битов, сообщение любой длины и параметр b, определяющий длину имитовставки кратный 64 битам, и выдает имитовставку заданной длины (рис. 1). В качестве основного алгоритма используется алгоритм CMAC, модифицированный следующим образом:

1) к входному сообщению добавляется 32 бита, содержащие длину этого сообщения;

2) если последний блок сообщения неполный, то последовательность 10..0 дописывается не в конец блока, как в CMAC, а в начало;

3) в качестве блочного шифра, работающего в режиме сцепления блоков шифротекста, используется не AES, а блочное преобразование E собственной разработки, описание которого приводится в п. 4.2;

4) блочное преобразование E получает на вход помимо результата сложения операцией XOR блока текста и результата шифрования предыдущего блока также порядковый номер входного блока (нумерация ведется сквозная по всему защищаемому сообщению).

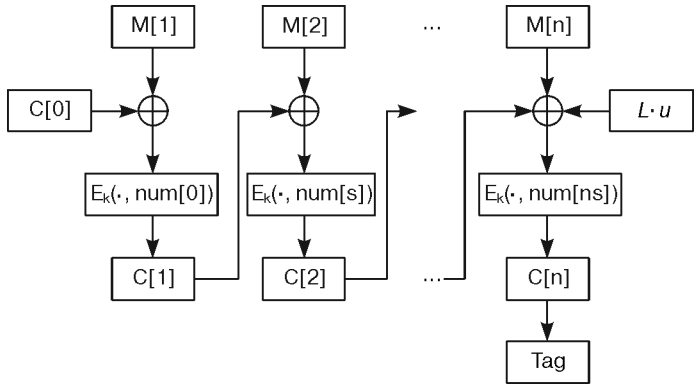

Рис. 1. Основной алгоритм кода аутентификации сообщения, когда последний блок полный

4.1.1. Генерация рабочих ключей K1, K2 из секретного ключа K

Для работы алгоритма из секретного ключа K вырабатываются два рабочих ключа по следующему алгоритму.

Вход: K — секретный ключ длины 256 битов.

Выход: K1, K2 — ключи длины b битов каждый.

1) $L = E_K(0^b, K)$;

2) $K1 = L \cdot u$, где выражение $L \cdot u$ представляет собой произведение L и u в поле GF(2^b);

3) $K2 = L \cdot u^2$.

На практике операция умножения в поле эффективно реализуется с помощью одной операции сдвига и одной операции XOR.

4.1.2. Основной алгоритм вычисления имитовставки

Вход: K — секретный ключ длины 256 битов;

b: $b = 64 \cdot s$ — длина имитовставки, которую необходимо вычислить, где $s \geq 1$;

M — входное сообщение.

Выход: Tag — имитовставка.

1) Пусть LenM — длина сообщения M в байтах. Дополняем M 32-битным блоком LenM.

$$M' = M \underbrace{\| LenM}_{32 \text{ бита}}.$$

2) Разбиваем M′ на блоки длины b битов (последний блок может быть неполным): $M' = M[1] \| M[2] \| \ldots \| M[n-1] \| M^*[n]$.

3) Если $M^*[n]$ — полный блок, то $M[n] = K1 \oplus M^*[n]$, иначе $M[n] = K2 \oplus$ $\oplus(10^j \| M^*[n])$, $j = nb - Mlen - 1$;

4) $C[0] = 0^b$;

5) num[0] = 0x9A3E15D38F5455C3;
6) Цикл по i от 1 до n;
   $C[i] = E_K(C[i-1] \oplus M[i], num[i-1]$;
7) Tag = C[n].

4.2. Алгоритм блочного преобразования E_K

На вход алгоритм получает блок длины равной длине имитовставки. Далее вычисляется вектор IV длины 64 бита, зависящий от всего блока. К каждому 64-битному блоку прибавляется операцией XOR полученный вектор IV и очередной элемент псевдослучайной последовательности (ведется «сквозное» вычисление псевдослучайной последовательности по всему сообщению). Результат шифруется по ГОСТ 28147—89 в режиме простой замены на секретном ключе K (рис. 2).

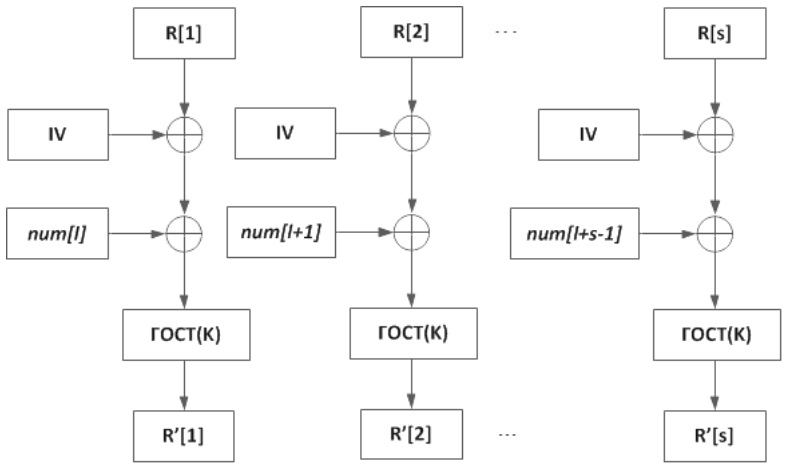

Рис. 2. Преобразование E_K

4.3. Алгоритм вычисления вектора IV

Цель вычисления IV — получить 64-битный вектор, каждый бит которого зависит от всего входного блока длины b битов. Для вычисления вектора IV предлагается три различных алгоритма:

1) на основе алгоритма PMAC (Parallelizable Message Authentication);
2) на основе линейного регистра сдвига над полем GF(2);
3) на основе линейного регистра сдвига над полем $GF(2^{64})$.

4.3.1. Алгоритм вычисления IV на основе алгоритма PMAC

Для вычисления IV используется алгоритм PMAC [10], в котором в качестве алгоритма блочного шифрования используется ГОСТ 28147—89 в режиме простой замены на секретном ключе K. Данный алгоритм позволяет распараллелить процесс обработки блоков входного текста, благодаря чему скорость работы алгоритма увеличивается.

4.3.2. Алгоритм вычисления IV на основе линейного регистра сдвига над полем GF(2)

Исходя из длины имитовставки, которую надо вычислить, определяется степень неприводимого многочлена над полем GF(2) $F(u) = u^b - \sum_{j=0}^{b-1} f_j \cdot u^j, \quad f_j \in GF(2)$. Данный многочлен является характеристическим многочленом линейного регистра сдвига (ЛРС).

На вход алгоритма подается вектор длины b битов, равной длине вырабатываемой имитовставки. Этот вектор является начальным заполнением ЛРС. Далее выполняется b + 63 тактов работы ЛРС. На *i*-м шаге вычисляется значение $v(i+b) = \sum_{j=0}^{b-1} f_j \cdot v(i+j)$, значения накопителей сдвигаются на 1 вправо, а $v(i+j)$ записывается в $(b-1)$ накопитель. Далее левые 64 бита заполнения ЛРС шифруются алгоритмом ГОСТ 28147—89 в режиме простой замены.

4.3.3. Алгоритм вычисления IV над полем GF(2^{64})

Предыдущий алгоритм показал весьма низкую скорость работы за счет вычисления большого числа тактов ЛРС в процедуре выработки IV. В данном алгоритме число раундов ЛРС сокращено с $(b + 63)$ до одного благодаря оперированию не с битами, а с 64-битными блоками.

Исходя из длины имитовставки, которую надо вычислить, определяется степень неприводимого многочлена над полем GF(2^{64}) $F(u) = u^{b/64} - \sum_{j=0}^{b/64-1} f_j \cdot u^j$, $f_j \in GF(2)$. Данный многочлен является характеристическим многочленом линейного регистра сдвига (ЛРС).

На вход алгоритма подается вектор длины b битов, равной длине вырабатываемой имитовставки. Данный вектор является начальным заполнением ЛРС. Далее выполняется 1 такт работы ЛРС. Вычисляется значение $v(b/64) = \sum_{j=0}^{b/64-1} f_j \cdot v(j)$, которое далее шифруются алгоритмом ГОСТ 28147—89 в режиме простой замены.

5. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ И ОЦЕНКА ЭФФЕКТИВНОСТИ

Предложенные три алгоритма были программно реализованы и протестированы [1—3]. Были проведены две серии экспериментов: первая — измерение времени вычисления имитовставок различной длины для файла фиксированной длины по всем алгоритмам, вторая — измерение времени вычисления имитовставок фиксированной длины (512 битов) для файлов различной длины. Вычисления проводились на персональном компьютере со следующими характеристиками: AMD Athlon(tm) XP 2600+ 1.91 ГГц, 512 Мб ОЗУ.

Зависимость времени вычисления от размера имитовставки

Для проведения экспериментальных исследований был выбран файл размером 2 Мб, для которого вычислялись имитовставки всевозможных длин (64, 128, 192, 256, 320, 384, 448, 512, 576, 640, 704, 768, 832, 896, 960, 1024 бит) по всем трем алгоритмам. Приведенный на рис. 3 график построен на основе полученных данных.

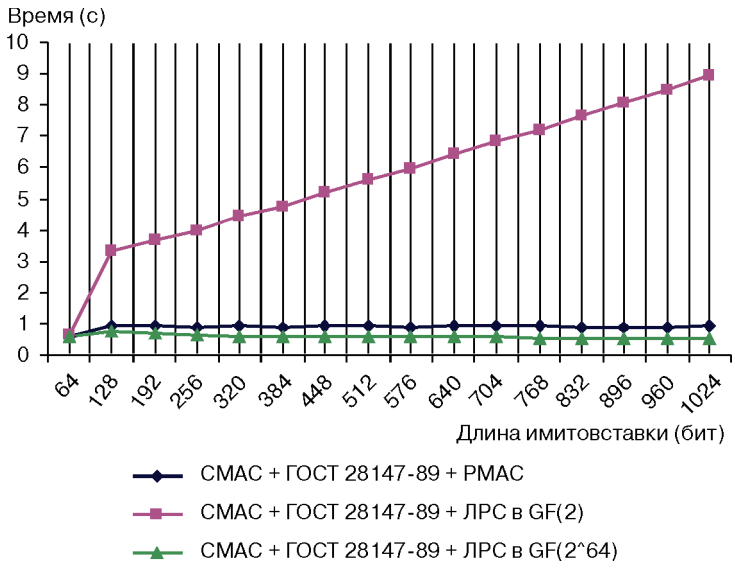

Рис. 3. Зависимость времени вычисления от длины имитовставки

Время вычислений по алгоритму, основанному на ГОСТ 28147—89, CMAC и ЛРС над GF(2), практически пропорционально увеличивается с ростом длины имитовставки. Данный эффект возникает в связи с тем, что при вычислении вектора IV для каждого блока данных длины b битов производится (b + 63) тактов ЛРС, где b — длина имитовставки в битах более 64 (для 64 битов IV не вычисляется). В связи с этим алгоритм неэффективно использовать для выработки имитовставки длины большей 64 битов. Однако имитовставку в 64 бита с такой же скоростью можно выработать и по остальным алгоритмам, показавшим эффективную работу и на более длинных имитовставках. Следовательно, использование алгоритма, основанного на ГОСТ 28147—89, CMAC и ЛРС над GF(2), не оправдано.

Алгоритм, основанный на ГОСТ 28147—89, CMAC и PMAC, и алгоритм, основанный на ГОСТ 28147—89, CMAC и ЛРС над $GF(2^{64})$, при увеличении длины имитовставки ведут себя схожим образом, однако первый несколько проигрывает второму в производительности начиная с 128 битов в связи с более сложным алгоритмом вычисления IV. Скорость вычислений для 128 битов почти в полтора раза меньше, чем для 64 битов, так как для вычисления 64-битной имитовставки фактически используется алгоритм блочного шифрования в режиме CBC без дополнитель-

ных преобразований. Для 128-битной имитовставки для каждого 128-битного блока данных вычисляется вектор IV, в связи с чем алгоритм теряет скорость.

Начиная с 128 битов наблюдается некоторое повышение скорости работы сдальнейшим увеличением длины имитовставки, так как сложность вычисления IV растет медленнее, чем рост длины имитовставки. Чем длиннее имитовставка, тем меньшее количество раз мы вычисляем IV, поэтому и возникает подобное явление.

С точки зрения скорости работы наиболее привлекательным выглядит алгоритм, основанный на ГОСТ 28147—89, CMAC и ЛРС над GF(2^{64}), однако и алгоритм, основанный на ГОСТ 28147—89, CMAC и PMAC, показывает приемлемый результат.

Зависимость времени вычисления от размера файла

Для проведения экспериментальных исследований были выбраны несколько файлов различной длины (2, 4, 6, 8, 10 Мб). Для каждого файла проводился сбор статистики времени вычислений имитовставок длины 512 битов для всех трех алгоритмов. На рис. 4 приведен график, построенный на основе полученных результатов.

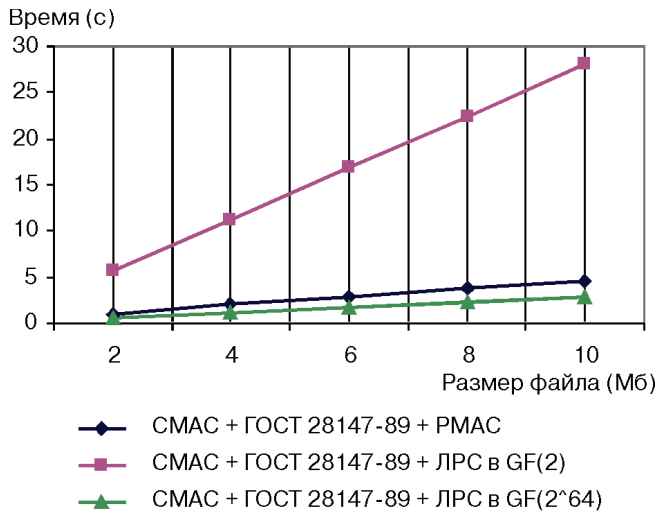

Рис. 4. Зависимость времени вычисления от размера входного файла

Время вычислений всех алгоритмов растет линейно с увеличением размера файла. Наилучший результат показал алгоритм, основанный на ГОСТ 28147—89, CMAC и ЛРС над GF(2^{64}), так как имеет более оптимальный алгоритм вычисления IV. Алгоритм, основанный на ГОСТ 28147—89, CMAC и ЛРС над GF(2), опять показал неудовлетворительный результат.

ВЫВОДЫ

Предложены три алгоритма кода аутентификации сообщений на основе ГОСТ 28147—89, позволяющие вычислять имитовставки любой длины, кратной 64 битам.

Работа всех трех алгоритмов была протестирована экспериментально с помощью программ, реализованных на языке C#. Наилучший результат показал алгоритм, использующий ЛРС над GF(264). Необходимо отдельно отметить свойство данного алгоритма увеличения скорости работы при увеличении длины вырабатываемой имитовставки. Данная особенность позволяет повышать уровень защищенности, увеличивая длину имитовставки, не теряя в производительности, а наоборот, увеличивая ее. Описанный алгоритм удовлетворяет современным требованиям, обладает большим запасом прочности и в перспективе может лечь в основу российского стандарта кода аутентификации сообщений.

ЛИТЕРАТУРА

[1] *Билык Т.А.* Свидетельство о государственной регистрации программы для ЭВМ № 2011616988 от 08.09.2011 «Код аутентификации сообщений на основе блочного преобразования, использующего PMAC» // Официальный бюллетень Российского агентства по патентам и товарным знакам «Программы для ЭВМ. Базы данных. Топологии интегральных микросхем». — 2011. — Вып. 4. — С. 443.

[2] *Билык Т.А.* Свидетельство о государственной регистрации программы для ЭВМ № 2011616990 от 08.09.2011 «Код аутентификации сообщений на основе блочного преобразования, использующего ЛРС над GF(2)» // Официальный бюллетень Российского агентства по патентам и товарным знакам «Программы для ЭВМ. Базы данных. Топологии интегральных микросхем». — 2011. — Вып. 4. — С. 444.

[3] *Билык Т.А.* Свидетельство о государственной регистрации программы для ЭВМ № 2011616989 от 08.09.2011 «Код аутентификации сообщений на основе блочного преобразования, использующего ЛРС над GF(2^64-83)» // Официальный бюллетень Российского агентства по патентам и товарным знакам «Программы для ЭВМ. Базы данных. Топологии интегральных микросхем». — 2011. — Вып. 4. — С. 443.

[4] *Билык Т.А.* Построение ключевой функции хэширования на основе отечественных криптографических алгоритмов // Материалы Ежегодной научно-технической конференции студентов, аспирантов и молодых специалистов МИЭМ. — М.: МИЭМ, 2009.

[5] *Билык Т.А.* Выработка требований к ключевой функции хэширования и разработки удовлетворяющего им алгоритма // Материалы Ежегодной научно-техническая конференция студентов, аспирантов и молодых специалистов МИЭМ. — М.: МИЭМ, 2010.

[6] *Билык Т.А., Дацун Н.Н., Потехонченко А.Ю.* Обеспечение безопасности передаваемых данных с помощью кодов аутентификации сообщений // Автоматизация в промышленности. — 2010. — № 12.

[7] *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.

[8] ГОСТ 28147—89. Системы обработки информации. Защита криптографическая.

[9] *Зубов А.Ю.* Математика кодов аутентификации. — М.: Гелиос АРВ, 2007.

[10] *Black J., Rogaway P.* A Block-Cipger Mode of Operation for Parallelizable Message Authentication. — URL: http://www.cs.ucdavis.edu/~rogaway/papers/pmac.pdf

[11] *Iwata T., Kurosawa K.* OMAC: One-Key CBC MAC. — URL: http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/omac/omac-spec.pdf

INFORMATION SECURITY WITH MESSAGE AUTHENTICATION CODE

T.A. Bilyk[1], A.A. Vnukov[2]

[1]Department of Information Security
Moscow State Institute of Electronics and Mathematics
B. Trehsvyatitelskiy per, 3/12, Moscow, Russia, 109028

[2]Department of Cybernetics and Mechatronics
Engineering Faculty
Peoples' Friendship University of Russia
Ordzhonikidze str., 3, Moscow, Russia, 115419

This article describes three Message authentication code algorithms. These algorithms ensure the authenticity of the message between two or more parties to the transaction and can be applied in such information bank systems as CFT and so on. Also the results of testing the software implementation of these algorithms are described here.

Key words: message authentication code, authenticity, integrity.