УДК 681.3.06

ИССЛЕДОВАНИЕ КОЛЛИЗИОННЫХ СВОЙСТВ КОДОВ АУТЕНТИФИКАЦИИ СООБЩЕНИЙ UMAC

А.А. КУЗНЕЦОВ, О.Г. КОРОЛЬ, С.П. ЕВСЕЕВ

Рассматривается алгоритм формирования кодов аутентификации сообщений UMAC, в основе которого лежит использование универсальных хеширующих функций. Предлагается уменьшенная модель UMAC (mini-UMAC) и методика статистического исследования коллизионных свойств формируемых кодов аутентификации сообщений. С использованием уменьшенной модели UMAC исследуются коллизионные свойства кодов аутентификации, показано, что применение криптографического преобразования (с использованием алгоритма AES) на завершающем этапе UMAC приводит к нарушению свойств универсального хеширования.

Ключевые слова: мини-UMAC, аутентификация, универсальное хеширование, коды аутентичности, алгоритм AES.

ПОСТАНОВКА ПРОБЛЕМЫ В ОБЩЕМ ВИДЕ И АНАЛИЗ ЛИТЕРАТУРЫ

Эффективным механизмом обеспечения целостности и аутентичности информации в современных телекоммуникационных системах и сетях является хеширование информации [1 – 7], применяемое как для формирования кодов обнаружения манипуляций (MDC – Manipulation Detection Code), так и для построения кодов аутентификации сообщений (MAC – Message Authentication Code) [5, 7].

Проведенный анализ показал, что наибольшей вычислительной эффективностью обладает отобранный при проведении европейского конкурса NESSIE алгоритм UMAC (Message Authentication Code using Universal Hashing) [5, 7], для формирования кодов аутентификации в котором используются семейства универсальных хеширующих функций [8, 9]. Число коллизий (столкновений) формируемых хеш-образов для каждого введенного ключа универсального хеширования не превышает некоторой заранее заданной величины, а криптостойкость UMAC обеспечивается на уровне выбранного криптоалгоритма (по спецификации рекомендован алгоритм шифрования AES). Однако влияние используемого криптоалгоритма на коллизионные свойства кодов подлинности сообщений UMAC на сегодняшний день не исследовано, обеспечение свойств универсального хеширования в такой многослойной конструкции не обосновано [1 – 7].

Целью данной работы является исследование коллизионных свойств хеширующих функций алгоритма UMAC, оценка влияния применяемого криптографического преобразования на последнем этапе формирования кодов аутентификации на обеспечение свойств универсального хеширования. Для этого в первой части статьи приводится общая конструкция схемы формирования кодов аутентификации сообщений UMAC, исследуются основные этапы (слои) преобразования для построения ключевых итерационных хеширующих функций. Во второй части статьи предлагается уменьшенная модель UMAC (mini-UMAC), позволяющая при сохранении математической структуры основных преобразований за счет уменьшения ключевого пространства и пространства аутентификаторов оценить число возникающих коллизий. Методика статистического исследования коллизионных свойств формируемых кодов аутентификации сообщений, с использованием уменьшенной модели UMAC, приводиться в третьей части статьи. Результаты моделирования и обсуждение полученных данных приводятся в четвертой части статьи, по которым делается вывод о нарушении свойств универсального хеширования UMAC.

1. ФОРМИРОВАНИЕ КОДОВ АУТЕНТИФИКАЦИИ СООБЩЕНИЙ С ИСПОЛЬЗОВАНИЕМ АЛГОРИТМА UMAC

Одна из первых версий алгоритма формирования кодов подлинности сообщений с использованием универсального хеширования (UMAC) была представлена в работе [1]. В дальнейшем, после некоторой доработки [2 – 4] алгоритм UMAC был представлен в финальном отчете европейского конкурса NESSIE – New European Schemes for Signatures, Integrity and Encryption (новые европейские схемы для подписей, целостности и шифрования) [5]. Одна из последних электронных версий алгоритма UMAC доступна в электронном виде [6]. Наиболее подробно отдельные компоненты UMAC изложены в диссертационной работе [7].

Рассмотрим общую схему формирования кодов подлинности сообщений с использованием алгоритма UMAC, проанализируем основные аналитические соотношения, описывающие внутреннюю структуру и применяемые преобразования при формировании кодов подлинности сообщений.

1.1. Общая схема формирования кодов подлинности сообщений с использованием алгоритма UMAC. Код подлинности сообщений (обозначим его *Tag*) по спецификации алгоритма

UMAC формируется посредством вычисления следующей функции:

$$Tag = UMAC(K, M, Nonce, Taglen) = Y \oplus Pad,$$

где: K – секретный ключ, длина которого $Keylen$ равна стандартной длине секретного ключа используемого блочного симметричного шифра (спецификацией UMAC рекомендуется использовать алгоритм шифрования AES (FIPS-197), в этом случае длина секретного ключа $Keylen$ принадлежит множеству допустимых значений $\{16, 24, 32\}$ байт); M – информационное сообщение, подлежащее аутентификации, представленное в виде массива-строки размерностью от одного до 2^{67} бит (2^{64} байт); $Nonce$ – неповторяющееся (для всех вводимых информационных сообщений M) восьмибайтное число; $Taglen$ – целое число из множества допустимых значений $\{4, 8, 12, 16\}$, задающее длину кода подлинности сообщений Tag в байтах; $Hash(K, M, Taglen)$ – функция ключевого универсального хеширования информационного сообщения M с использованием секретного ключа K; $PDF(K, Nonce, Taglen)$ – функция формирования псевдослучайной подложки (Pad) по введенному значению $Nonce$ и секретному ключу K; «$\oplus$» – побитовое сложение (XOR) результата ключевого хеширования сообщения $Y = Hash(K, M, Taglen)$ и сформированной подложки

$$Pad = PDF(K, Nonce, Taglen), \text{ т.е.}$$

$$Tag = Hash(K, M, Taglen) \oplus PDF(K, Nonce, Taglen).$$

Длина хеш-кода Y, подложки Pad и кода Tag принадлежат множеству допустимых значений $\{32, 64, 96, 128\}$ бит. Эти фиксированные значения $Taglen$ соответствуют случаю формирования кодов подлинности сообщений UMAC – 32, UMAC – 64, UMAC – 96 или UMAC – 128, соответственно.

Рассмотрим схему формирования хеш-кодов $Y = Hash(K, M, Taglen)$ и подложки $Pad = PDF(K, Nonce)$.

1.2. Схема формирования хеш-кодов $Y = Hash(K, M, Taglen)$. Вычисление значения функции $Hash(K, M, Taglen)$ ключевого универсального хеширования информационного сообщения M с использованием секретного ключа K выполняется в три этапа (используется три уровня (слоя) ключевого хеширования) $Hash_{L1}$, $Hash_{L2}$ и $Hash_{L3}$, соответственно. Второй уровень хеширования $Hash_{L2}$ выполняется только если длина хешируемого сообщения M превосходит 1024 байт.

Длина хеш-кода Y кратна 32 битам, его значение $Y = Hash(K, M, Taglen)$ для любой длины $Taglen$ формируется посредством объединения (конкатенации) нескольких (от одной до четырех) последовательностей Y_{L3_i}

$$Y = Hash(K, M, Taglen) = Y_{L3_1} \| Y_{L3_2} \| ... \| Y_{L3_{It}},$$

$$It = Taglen / 4,$$

где Y_{L3_i} – результат многоуровневого хеширования сообщения M на i-ой итерации с использованием соответствующих ключей, $i = 1, 2, ..., It$.

Рассмотрим формирование хеш-кода Y_{L3_i} на i-ой итерации. Для этого обозначим результат многоуровневого хеширования на произвольной i-ой итерации следующим образом:

$$Y_{L3_i} = Y_{L3} =$$

$$= Hash_{L3}\left(K_{L3_1}, K_{L3_2}, Hash_{L2}\left(K_{L2}, Hash_{L1}(K_{L1}, M)\right)\right),$$

где: $Hash_{L1}(K_{L1}, M)$, $Hash_{L2}(K_{L2}, Y_{L1})$ и $Hash_{L3}\left(K_{L3_1}, K_{L3_2}, Y_{L2}\right)$ – функции ключевого хеширования первого, второго и третьего уровня, управляемые и зависящими от номера итерации секретными ключами K_{L1}, K_{L2}, K_{L3_1}, K_{L3_2}, соответственно.

Ключевые последовательности K_{L1}, K_{L2}, K_{L3_1}, K_{L3_2} формируются по введенному секретному ключу K длины $Keylen$ байт с использованием специальной функции $KDF(K, Index, Numbyte)$ (Key-Derivation Function – KDF), где $Index$ и $Numbyte$ представляют собой два целых положительных числа, не превосходящих 2^{64}.

Первый уровень хеширования выполняет разбиение массива-строки M размерности до 2^{64} байт на блоки M_i по 1024 байт с последующим преобразованием каждого блока функцией $NH(K_{L1}, M_i)$. Полученные результаты $Hash_{L1_i} = NH(K_{L1}, M_i)$ конкатенируются (объединяются) в строку $Y_{L1} = Hash_{L1}(K_{L1}, M)$, которая короче информационной последовательности в 128 раз. Эта строка и является результатом хеширования первого уровня:

$$Y_{L1} = Hash_{L1}(K_{L1}, M) =$$

$$= NH(K_{L1}, M_0) \| NH(K_{L1}, M_1) \| ... \| NH(K_{L1}, M_{n-1}),$$

где $n = \left\lceil \frac{Length(M)}{1024} \right\rceil$, $\lceil x \rceil$ – целая часть числа x, $Length(M)$ – байтовая длина информационного сообщения M.

Значение функции $Hash_{L1_i} = NH(K_{L1}, M_i)$ вычисляется по следующему правилу. Информационный блок M_i разбивается на четырехбайтовые подблоки так, что

$$M_i = M_{i_1} \| M_{i_2} \| ... \| M_{i_t},$$

где $t = \left\lceil \frac{Length(M_i)}{4} \right\rceil$. В данном случае

$$t = \left\lceil \frac{1024}{4} \right\rceil = 256.$$

Аналогичным образом ключевая последовательность K_{L1} представляется в виде последовательностей четырехбайтовых подблоков:

$$K_{L1} = K_{L1_1} \| K_{L1_2} \| ... \| K_{L1_t}.$$

После чего (принимая начальное состояние $Hash_{L1_i} = 0$) для всех $j = 1,9,17...,t-7$ выполняются следующие операции:

$$Hash_{L1_i} = Hash_{L1_i} +$$
$$+_{64}((M_{i_{j+0}} +_{32} K_{L1_{j+0}}) \times_{64} (M_{i_{j+4}} +_{32} K_{L1_{j+4}})),$$
$$Hash_{L1_i} = Hash_{L1_i} +$$
$$+_{64}((M_{i_{j+1}} +_{32} K_{L1_{j+1}}) \times_{64} (M_{i_{j+5}} +_{32} K_{L1_{j+5}})),$$
$$Hash_{L1_i} = Hash_{L1_i} +$$
$$+_{64}((M_{i_{j+2}} +_{32} K_{L1_{j+2}}) \times_{64} (M_{i_{j+6}} +_{32} K_{L1_{j+6}})),$$
$$Hash_{L1_i} = Hash_{L1_i} +$$
$$+_{64}((M_{i_{j+3}} +_{32} K_{L1_{j+3}}) \times_{64} (M_{i_{j+7}} +_{32} K_{L1_{j+7}})),$$

где $+_{64}$, $+_{32}$ – операции сложения по модулю 2^{64} и 2^{32}, соответственно; $\times_{64}$ – операция умножения по модулю 2^{64}.

В работах [1 – 7] показано, что рассмотренная функция ключевого хеширования *NH* принадлежит к классу универсальных хеширующих функций.

Второй уровень хеширования использует полиномиальное ключевое хеширование *Poly*, подробно рассмотренное в работах [1 – 7]. Результатом работы этого уровня есть вычисление хеш-кода

$$Y_{L2} = Hash_{L2}(K_{L2}, Y_{L1}) =$$
$$= Poly(Wordbits, Maxwordrange, k, M_P),$$

т.е. на вход хеширования второго уровня подается строка $Y_{L1} = Hash_{L1}(K_{L1}, M)$.

В качестве исходных данных функция полиномиального хеширования использует:

$Wordbits \in [64,128]$; *Maxwordrange* – положительное целое число, меньшее $2^{Wordbits}$; k – зависящее от ключа K_{L2} целое число из диапазона $[0,..., prime(Wordbits)-1]$, $prime(x)$ – наибольшее простое число, меньшее 2^x;

$M_P = Y_{L1} = Hash_{L1}(K_{L1}, M)$ – данные, подлежащие полиномиальному хешированию.

По спецификации алгоритма UMAC в качестве $prime(x)$ используются следующие константы: $prime(36) = 2^{36} - 5$, $prime(64) = 2^{64} - 59$, $prime(128) = 2^{128} - 159$. Битовую длину M_P обозначим $Bytelength(M_P)$. В зависимости от длины M_P используются следующие особенности в реализации второго уровня хеширования:

– если длина поступивших данных M_P не превосходит 2^{17} байт, тогда полиномиальное хеширование *Poly* выполняется с параметрами $Wordbits = 64$; $Maxwordrange = 2^{64} - 2^{32}$; $k = k64$ – строка, образованная первыми восемью байтами ключа K_{L2} и специальной восьмибайтной маской;

– если длина поступивших данных M_P превосходит 2^{17} байт (но не превосходит 2^{64} байт), тогда первые 2^{17} байт данных обрабатываются функцией полиномиального хеширования $Poly(64, 2^{64} - 2^{32}, k64, M_P)$, а оставшиеся байты данных обрабатываются функцией *Poly* с параметрами $Wordbits = 128$; $Maxwordrange = 2^{128} - 2^{96}$; $k = k128$ – строка, образованная последними 16 байтами ключа K_{L2} и специальной 16 байтной маской.

Хешируемые данные M_P разбиваются на блоки по $Wordbytes = Wordbits / 8$ байт:

$$M_P = M_{P_1} \| M_{P_2} \| ... \| M_{P_n},$$

где $n = Bytelength(M_P) / Wordbytes$.

Результатом хеширования является значение полиномиальной функции

$$Y_{L2} = \left(M_{P_n} + kM_{P_{n-1}} + ... + k^{n-1} M_{P_1} + k^n\right) \bmod(p),$$

которое вычисляется итеративной процедурой (для всех $i = 1,2,...,n$):

$$Poly_i = \left(kPoly_{i-1} + M_{P_i}\right) \bmod(p), \quad Poly_0 = 1,$$
$$p = prime(Wordbits)$$

с помощью схемы Горнера

$$M_{P_n} + kM_{P_{n-1}} + ... + k^{n-1} M_{P_1} + k^n =$$
$$= (((k + M_{P_1})k + M_{P_2})k + ... + M_{P_{n-1}})k + M_{P_n}.$$

Вычисленное хеш-значение $Y_{L2} = Poly_n$ является целым числом из диапазона

$$[0,..., prime(Wordbits) - 1].$$

Рассмотренная функция полиномиального ключевого хеширования

$$Poly(Wordbits, Maxwordrange, k, M_P)$$

принадлежит к классу универсальных хеширующих функций [1 – 7].

Третий уровень хеширования

$$Hash_{L3}\left(K_{L3_1}, K_{L3_2}, Y_{L2}\right)$$

выполняется над результатом полиномиального хеширования и преобразует поданные на его вход данные длины до 16 байт в хеш-код Y фиксированной длины 32 бита.

В качестве исходных данных третьего уровня хеширования выступают две ключевых последовательности K_{L3_1} и K_{L3_2} длины 64 и 4 байта соответственно, а также входная 16 байтная последовательность Y_{L2}.

Хешируемые данные Y_{L2} и ключевая последовательность K_{L3_1} равномерно разбиваются на восемь блоков, каждый из которых представляется как целое число Y_{L2_i} и $K_{L3_{1i}}$, $i = 1,2,...,8$.

Хеш-значение Y_{L3} вычисляется следующим образом:

$$Y_{L3} =$$
$$= \left(\left(\left(\sum_{i=1}^{m} Y_{L2_i} K_{L3_{1i}}\right) \bmod(prime(36))\right) \bmod(2^{32})\right) xor(K_{L3_2}),$$

где $(x)xor(y)$ – операция «исключающего ИЛИ» над значениями x и y.

Рассмотренная функция ключевого хеширования $Y_{L3} = Hash_{L3}\left(K_{L3_1}, K_{L3_2}, Y_{L2}\right)$ принадлежит к

классу универсальных хеширующих функций, ее свойства подробно исследованы в работах [1–7].

1.3. Схема формирования ключей (KDF: Key-Derivation Function). Специальная функция $KDF(K, Index, Numbyte)$ предназначена для формирования последовательностей псевдослучайных бит данных, которые используются на различных уровнях формирования кодов подлинности сообщений как ключевые данные соответствующих функций хеширования.

В качестве исходных данных функции генерации ключевых псевдослучайных последовательностей используется секретных ключ K длины $Keylen$ байт и два положительных целых числа $Index$ и $Numbyte$, значение которых не превосходит 2^{64}.

Для формирования псевдослучайных ключевых последовательностей используется блочный симметричный шифр. Обозначим процедуру шифрования блока данных T длины $Blocklen$ байт с использованием секретного ключа K длины $Keylen$ байт в виде некоторой функции $Enchiper(K,T)$. Тогда процедуру формирования псевдослучайной ключевой последовательности $K' = KDF(K, Index, Numbyte)$ можно представить в виде следующего итеративного (для всех $i = 1,2,...,n$) преобразования:

$$T_i = Index \| i,$$

$$K'_i = Enchiper(K, T_i),$$

$$K' = K'_1 \| K'_2 \| ... \| K'_n,$$

где $n = \left\lceil \frac{Numbyte}{Blocklen} \right\rceil$, $\lceil x \rceil$ – целая часть числа x, $a \| b$ – конкатенация (присоединение) строк a и b.

Сформированная последовательность псевдослучайных ключевых бит данных K' имеет длину $Numbyte$ байт, кратную длине блока $Blocklen$ байт.

1.4. Схема формирования псевдослучайной подложки (PDF: Pad-Derivation Function). Функция $PDF(K, Nonce, Taglen)$ предназначена для формирования псевдослучайной подложки Pad, используемой на заключительном этапе формирования кода подлинности сообщения.

В качестве исходных данных используется секретных ключ K длины $Keylen$ байт и неповторяющееся (для всех вводимых информационных сообщений M) восьмибайтное число $Nonce$, а также целое число $Taglen$, задающее размер (длину в байтах) формируемого кода подлинности Tag.

Процедура формирования псевдослучайной подложки $Pad = PDF(K, Nonce, Taglen)$ состоит в формировании подключа

$$K' = KDF(K, Index, Numbyte),\ Index = 0,$$

$$Numbyte = Keylen,$$

с использованием рассмотренной выше процедуры формирования последовательностей псевдослучайных ключевых бит и шифрования значения $Nonce$ на сформированном подключе K', т.е.:

$$Pad = PDF(K, Nonce, Taglen) =$$

$$= Enchiper(KDF(K, 0, Keylen), Nonce).$$

Процедура формирования псевдослучайной подложки Pad построена так, что результирующее значение Pad имеет длину $Taglen$ байт вне зависимости от значений $Blocklen$ и $Nonce$.

Таким образом, рассмотренная схема формирования кодов подлинности сообщений UMAC использует многоуровневую конструкцию универсального хеширования $Hash(K, M, Taglen)$ и процедуру формирования псевдослучайной подложки Pad. Применение универсального хеширования позволяет обеспечить равновероятность формирования хеш-образов для всего множества используемых ключевых данных, на чем и базируется доказательство безопасности алгоритма [1–7]. Формирование псевдослучайной подложки криптографически стойким алгоритмом (например, с использованием блочного симметричного шифра AES) обеспечивает криптостойкость алгоритма UMAC на уровне стойкости применяемого криптоалгоритма [5, 7]. Следовательно, рассмотренная схема формирования UMAC обладает потенциально высокими показателями эффективности.

В тоже время на сегодняшний день не исследованы коллизионные свойства алгоритма UMAC после применения завершающей процедуры наложения на формируемые хеш-коды $Y = Hash(K, M, Taglen)$ псевдослучайных подложек $Pad = PDF(K, Nonce, Taglen)$. Ниже показано, что результирующие коды подлинности сообщений $Tag = UMAC(K, M, Nonce, Taglen) = Y \oplus Pad$ формируются не равновероятно для всего множества используемых ключевых данных. Следовательно алгоритм формирования UMAC после применения последнего слоя наложения псевдослучайных подложек теряет свойство «универсальности» хеширования его коллизионные свойства существенно ухудшаются.

Для проведения исследований коллизионных свойств кодов аутентификации сообщений, сформированных по рассмотренной выше схеме, предлагается использовать уменьшенную модель UMAC (mini-UMAC). Применение уменьшенных моделей позволяет, сохранив алгебраическую структуру криптоалгоритма, исследовать основные показатели его эффективности [10 – 14]. Этот поход широко используется на сегодняшний день при исследовании криптографических свойств блочных симметричных шифров. Так, например, в работах [12 – 14] разработаны уменьшенные модели криптоалгоритмов AES, Camelia, ADE, Лабиринт, Калина, Мухомор и др., использование которых позволило экспериментально исследовать дифференциальные и

линейные свойства соответствующих шифров, оценить их устойчивость к атакам дифференциального и линейного криптоанализа. Коме того, на основе анализа уменьшенных моделей в работах [12 – 14] предложен подход к оценке эффективности блочных симметричных шифров в виде вычислительных затрат, требуемых для достижения шифром асимптотических характеристик случайной подстановки.

В настоящей работе предлагается дальнейшее развитие данного направления, состоящее в использовании уменьшенных моделей отдельных слоев преобразований для оценки коллизионных свойств, формируемых кодов аутентификации сообщений.

2. УМЕНЬШЕННАЯ МОДЕЛЬ UMAC (MINI-UMAC)

Схема формирования кодов аутентификации сообщений UMAC использует в своей структуре несколько слоев преобразования, в том числе блочный симметричных шифр (рекомендован к использованию шифр AES). Разрабатываемая уменьшенная модель UMAC должна включать соответствующие слои преобразования с сохранением их алгебраической структуры при выполнении масштабирования до мини-версии. Естественным представляется исследовать коллизионные характеристики формируемых образов (кодов) на каждом из слоев преобразования, в том числе формируемых с помощью блочного симметричного шифра псевдослучайных подложек Pad, проанализировать их влияние на коллизионные свойства в целом, т.е. на коллизионные свойства кодов аутентификации сообщений уменьшенной модели UMAC.

Выше было показано, что схема формирования кодов UMAC состоит из следующих слоев:

– трехуровневое универсальное хеширование для формирования хеш-кодов

$$Y = Hash(K, M, Taglen);$$

– криптографическое преобразования с использованием блочного симметричного шифра для формирования псевдослучайной подложки

$$Pad = PDF(K, Nonce, Taglen);$$

– заключительное преобразование для формирования кодов аутентификации сообщений

$$Tag = UMAC(K, M, Nonce, Taglen) = Y \oplus Pad .$$

Рассмотрим каждый слой схемы формирования кодов аутентификации сообщений UMAC на предмет их масштабирования.

2.1. Мини-версию трехуровневого универсального хеширования построим без изменения структуры алгебраических преобразований простым уменьшением размерности блоков обрабатываемых данных в восемь раз.

Соответствующая длина хеш-кода $Y_{\min i}$ уменьшенной модели первого слоя будет кратна 4 битам, его значение сформируем посредством объединения (конкатенации) четырех последовательностей $Y_{\min iL3_i}$

$$Y_{\min i} = Y_{\min iL3_1} \| Y_{\min iL3_2} \| Y_{\min iL3_3} \| Y_{\min iL3_4} ,$$

где $Y_{\min iL3_i}$ – результат многоуровневого хеширования сообщения уменьшенной модели первого слоя mini-UMAC.

Рассмотрим процесс формирования одного блока $Y_{\min iL3_i}$ (второй уровень хеширования в уменьшенной модели выполнять не будем):

$$Y_{\min iL3_i} = Y_{\min iL3} =$$

$$= Hash_{\min iL3}\left(K_{\min iL3_1}, K_{\min iL3_2}, Hash_{\min iL1}\left(K_{\min iL1}, M_{\min i}\right)\right),$$

где $K_{\min iL1}$, $K_{\min iL3_1}$, $K_{\min iL3_2}$ – ключевые последовательности mini-UMAC, $Hash_{\min iL1}$ и $Hash_{\min iL3}$ – уменьшенные версии хеширования первого и третьего уровней соответственно.

На первом уровне массив-строка $M_{\min i}$ размерности 32 бита преобразуется функцией $NH(K_{L1}, M_i)$. Эта строка и является результатом хеширования первого уровня: $Y_{\min iL1} = NH_{\min i}(K_{\min iL1}, M_{\min i})$.

Значение функции $NH_{\min i}(K_{\min iL1}, M_{\min i})$ вычисляется по следующему правилу. Информационный блок $M_{\min i}$ разбивается на восемь четырехбитовых подблоков

$$M_{\min i} = M_{\min i_1} \| M_{\min i_2} \| ... \| M_{\min i_8} .$$

Аналогичным образом ключевая последовательность K_{L1} представляется в виде последовательностей из восьми четырехбитовых подблоков: $K_{\min iL1} = K_{\min iL1_1} \| K_{\min iL1_2} \| ... \| K_{\min iL1_8}$.

После чего (принимая начальное состояние $Hash_{L1} = 0$) выполняются следующие операции:

$$Hash_{\min iL1} = Hash_{\min iL1} +$$
$$+_8((M_{\min i_0} +_4 K_{\min iL1_0}) \times_8 (M_{\min i_4} +_4 K_{\min iL1_4})),$$

$$Hash_{\min iL1} = Hash_{\min iL1} +$$
$$+_8((M_{\min i_1} +_4 K_{\min iL1_1}) \times_8 (M_{\min i_5} +_4 K_{\min iL1_5})),$$

$$Hash_{\min iL1} = Hash_{\min iL1} +$$
$$+_8((M_{\min i_2} +_4 K_{\min iL1_2}) \times_8 (M_{\min i_6} +_4 K_{\min iL1_6})),$$

$$Hash_{\min iL1} = Hash_{\min iL1} +$$
$$+_8((M_{\min i_3} +_4 K_{\min iL1_3}) \times_8 (M_{\min i_7} +_4 K_{\min iL1_7})),$$

где $+_8$, $+_4$ – операции сложения по модулю 2^8 и 2^4, соответственно; $\times_8$ – операция умножения по модулю 2^8.

В результате вычислений формируется восьмибитное значение $Y_{\min iL1} = Hash_{\min iL1}$.

Третий уровень хеширования преобразует поданные на его вход восьмибитные данные $Y_{\min iL1}$ в хеш-код $Y_{\min iL3}$ длины 4 бита. В качестве ключевых последовательностей выступают $K_{\min iL3_1}$ и $K_{\min iL3_2}$ длины 16 и 4 бита соответственно.

Хешируемые данные $Hash_{\min iL1}$ и ключевая последовательность $K_{\min iL3_1}$ равномерно разбиваются на четыре блока, каждый из которых представляется как целое число $Y_{\min iL2_i}$ и $K_{\min iL3_{1i}}$, $i=1,2,...,4$.

Хеш-значение $Y_{\min iL3}$ вычисляется следующим образом:

$$Y_{\min iL3} =$$

$$=\left(\left(\left(\sum_{i=1}^{4} Y_{\min iL2_i} K_{\min iL3_{1i}}\right)\bmod(17)\right)\bmod(2^4)\right) xor(K_{\min iL3_2}),$$

где $(x)xor(y)$ – операция «исключающего ИЛИ» над значениями x и y.

2.2. Мини-версия блочного симметричного шифра AES для формирования псевдослучайной подложки подробно рассмотрена в работах [10-14]. Наиболее простой в реализации есть мини-версия шифра AES (Baby-Rijndael), которая предложена К. Бергманом [10]. Кратко рассмотрим эту уменьшенную модель шифра и обоснуем ее использование для формирования псевдослучайной подложки в mini-UMAC.

Размер блока открытого текста равен 16 бит, которые обозначим четырьмя шестнадцатеричными числами h_0, h_1, h_2, h_3. Отметим, что h_0 состоит из первых четырех бит входного потока. Однако когда h_0 рассматривается как шестнадцатеричная цифра, первый бит рассматривается, как бит высшего порядка. Например, входной блок 1000 1100 0111 0001 будет представлен $h_0=8$, $h_1=c$, $h_2=7$, $h_3=1$.

Размер ключа также равен 16 бит. Обозначим его как 4 шестнадцатеричных чисел k_0, k_1, k_2, k_3.

Шаги шифра применяются к состоянию – массиву 2×2 шестнадцатеричных цифр. Однако для рассматриваемой ниже операции $\tilde{\sigma}$ состояние будет представлено как массив 8×2 бит, т.е. каждая шестнадцатеричная цифра будет, рассматривается как столбец 4 бит с битом высшего порядка сверху.

Входной блок загружается в состояние отображениемv h_0, h_1, h_2, h_3 в $\begin{bmatrix} h_0 & h_2 \\ h_1 & h_3 \end{bmatrix}$. Например, входной блок 1000 1100 0111 0001 будет загружен как $\begin{bmatrix} 8 & 7 \\ c & 1 \end{bmatrix}$, где матрица 8×2 будет $\begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$.

Baby-Rijndael включает несколько идентичных по структуре раундов (по умолчанию их 4). Перед шифрованием входной блок загружается в состояние, как описано выше и рассчитываются раундовые ключи. Шифрование имеет общую структуру:

$$E(a)=r_4\circ r_3\circ r_2\circ r_1\circ(a\oplus k_0),$$

где a обозначает состояние, k_0, k_1, k_2, k_3, k_4 – раундовые ключи и $r_i(a)=(t\cdot\tilde{\sigma}(S(a)))\oplus k_i$, за исключением r_4, где пропущено умножение на t. В конце шифра состояние сгружается в 16-битный блок в таком же порядке, в котором он загружался.

Теперь опишем отдельные компоненты шифра.

SubBytes: Операция S есть выборочная таблица, которая применяется к каждой 16-ричной цифре состояния:

$$\begin{bmatrix} h_0 & h_2 \\ h_1 & h_3 \end{bmatrix} \xrightarrow{S} \begin{bmatrix} S(h_0) & S(h_2) \\ S(h_1) & S(h_3) \end{bmatrix},$$

где функция S задается следующей таблицей 1.

Таблица 1

Выборочная таблица, реализующая S-блок Baby-Rijndael

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$S(x)$	a	4	3	b	8	e	2	c	5	7	6	f	0	1	9	d

ShiftRows: Операция $\tilde{\sigma}$ просто меняет входы во второй строке состояния:

$$\begin{bmatrix} h_0 & h_2 \\ h_1 & h_3 \end{bmatrix} \xrightarrow{\tilde{\sigma}} \begin{bmatrix} h_0 & h_2 \\ h_3 & h_1 \end{bmatrix}.$$

MixColumns: Матрица t является следующей 8Ч8 матрицей бит:

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Для этого преобразования состояние рассматривается как 8×2 битовая матрица. Состояние умножается слева на t, используя матричное умножение по модулю 2: $a=ta$.

KeySchedule: В начале шифра и в конце каждого раунда состояние побитно складывается (т.е. по модулю 2) с раундовым ключом. Столбцы раундовых ключей определены рекурсивно следующим образом:

$$w_0=\begin{pmatrix} k_0 \\ k_1 \end{pmatrix},\ w_1=\begin{pmatrix} k_2 \\ k_3 \end{pmatrix},$$

$$w_{2i}=w_{2i-2}\oplus S(reverse(w_{2i-2}))\oplus r_i,$$

$$\{w_{2i+1}=w_{2i-1}\oplus w_{2i}\}$$

для всех $i=1,2,3,4$, где $r_i=\begin{pmatrix} 2^{i-1} \\ 0 \end{pmatrix}$, а функция *reverse* заменяет два входа в столбец. Функция S та же, что и описанная выше.

Следует заметить, что все сложения выполняются побитно по модулю 2. Наконец, для

$i = 1,2,3,4$ раундовый ключ k_i есть матрица, чьи столбцы есть w_{2i} и w_{2i+1}.

Использование рассмотренной уменьшенной модели блочного симметричного шифра AES позволяет провести экспериментальные исследования коллизионных свойств формируемых псевдослучайных подложек по всему множеству секретных ключей. Так, псевдослучайная подложка $Pad_{\min i}$ mini-UMAC формируется посредством шифрования неповторяющегося для каждого информационного сообщения $M_{\min i}$ числа *Nonce*. Результирующее значение $Pad_{\min i}$ имеет длину 16 бит, так же, как и соответствующая длина хеш-кода $Y_{\min i}$.

2.3. Мини-версия заключительного преобразования для формирования кодов аутентификации сообщений mini-UMAC состоит в поразрядном суммировании по модулю 2 значений $Y_{\min i}$ и $Pad_{\min i}$: $Tag_{\min i} = Y_{\min i} \oplus Pad_{\min i}$.

Таким образом, масштабирование применяемых преобразований на соответствующих слоях схемы формирования кодов аутентификации сообщений, позволяет построить уменьшенную модель UMAC, экспериментально исследовать коллизионные свойства формируемых образов (кодов). Коэффициент масштабирования при разработке мини-модели UMAC выбран таким образом, чтобы длина формируемых хеш-кодов Y, псевдослучайных подложек Pad и кодов аутентификации сообщений $Tag = Y \oplus Pad$ была равна длине блока мини-версии блочного симметричного шифра AES [10], т.е. 16 битам. Выбор такого коэффициента масштабирования позволяет с одной стороны сохранить алгебраическую структуру основных преобразований алгоритма UMAC, в том числе и входящего в его схему алгоритма AES, с другой стороны это дает возможность провести экспериментальные исследования с использованием методов статистической проверки гипотез и математической статистики, рассматривая ограниченный набор элементов Y, Pad и $Tag = Y \oplus Pad$ и соответствующие результаты по оценке числа коллизий как выборку из генеральной совокупности.

Обоснуем методику статистического исследования коллизионных свойств формируемых элементов (обозначим их для простоты $h(x)$), рассмотрим основные условия и ограничения при проведении экспериментов.

3. МЕТОДИКА СТАТИСТИЧЕСКОГО ИССЛЕДОВАНИЯ КОЛЛИЗИОННЫХ СВОЙСТВ

Проведение экспериментальных исследований коллизионных свойств кодов аутентификации сообщений UMAC проведем по соответствующим слоям преобразования:

1. На первом этапе исследуем коллизионные свойства мини-версии универсального хеширования. Для этого необходимо подтвердить в ходе эксперимента теоретические оценки числа возникающих коллизий формируемых хеш-кодов $Y_{\min i}$;

2. На втором этапе проведем экспериментальные исследования коллизионных свойств псевдослучайных подложек $Pad_{\min i}$ на основе анализа свойств уменьшенной модели шифра Baby-Rijndael. Подобные исследования в доступной литературе не описаны и, по всей видимости, проводятся нами впервые;

3. На третьем этапе проведем экспериментальные исследования коллизионных свойств формируемых с использованием mini-UMAC кодов аутентификации сообщений $Tag_{\min i} = Y_{\min i} \oplus Pad_{\min i}$. Это наиболее важная часть проводимых исследований, поскольку она позволит ответить на вопрос о сохранении свойств универсального хеширования после применения слоя криптографического преобразования информации.

Оценку числа коллизий формируемых элементов будем проводить, ориентируясь на коллизионные свойства универсального хеширования. Собственно говоря, нам требуется подтвердить или опровергнуть гипотезу о сохранении коллизионных свойств универсального хеширования на всех этапах формирования кодов аутентификации сообщений mini-UMAC.

Идея универсального хеширования заключается в определении такого набора элементов конечного множества H хеш-функций $h: A \to B$, $|A| = a$, $|B| = b$ чтобы случайный выбор функции $h \in H$ обеспечивал бы низкую вероятность коллизии, т.е. для любых различных входов x_1 и x_2 вероятность того, что $h(x_1) = h(x_2)$ (вероятность коллизии, столкновения) не должна превосходить некоторой заранее заданной величины ε:

$$P_{\text{кол}} = P(h(x_1) = h(x_2)) \le \varepsilon,$$

причем вероятность коллизии может быть рассчитана как

$$P_{\text{кол}} = \frac{\delta_H(x_1, x_2)}{|H|},$$

где $\delta_H(x_1, x_2)$ – количество таких хеш-функций в H, при которых значения $x_1, x_2 \in A$, $x_1 \ne x_2$ вызывают коллизию, т.е. $h(x_1) = h(x_2)$.

Приведем два определения универсального хеширования [8, 9].

1. Пусть $0 < \varepsilon < 1$. H является ε – *универсальным хеш-классом (сокращенно* $\varepsilon - U(H, A, B)$*),* если для двух различных элементов $x_1, x_2 \in A$ существует не больше, чем $|H| \cdot \varepsilon$ функций $f \in H$ таких, что $h(x_1) = h(x_2)$, если $\delta_H(x_1, x_2) \le \varepsilon |H|$ для всех $x_1, x_2 \in A$, $x_1 \ne x_2$.

2. Пусть $0 < \varepsilon < 1$. H является ε – *строго универсальным хеш-классом* (сокращенно $\varepsilon - SU(H, A, B)$) если выполняются следующие условия:

– для каждого $x_1 \in A$ и для каждого $y_1 \in B$,

$$\left|\{h \in H : h(x_1) = y_1\}\right| = |H| / |B|;$$

– для каждого $x_1, x_2 \in A$, $x_1 \neq x_2$ и для каждого $y_1, y_2 \in B$,

$$\left|\{h \in H : h(x_1) = y_1, h(x_2) = y_2\}\right| \leq \varepsilon |H| .$$

Определение универсального класса хеш-функций эквивалентно определению такого алгоритма формирования кода аутентификации, при котором число различных правил формирования кода аутентификации (число ключей), при которых существует коллизия (совпадение кодов аутентификации) для двух произвольных входных последовательностей, ограничено. Число таких ключей не может превосходить значение $P_{\text{кол}} \cdot |H|$, где $P_{\text{кол}}$ – вероятность коллизии, $|H|$ – число всех правил (ключей).

Определение строго универсального класса хеш-функций эквивалентно определению такого алгоритма формирования кодов аутентификации, при котором будут выполняться следующие условия:

1. Число правил формирования кода аутентификации (число ключей), при которых для произвольной входной последовательности значение кода аутентификации не изменяется, ограничено. Число таких ключей не может превосходить значения $|H|/|B|$, где $|H|$ – число всех ключей, $|B|$ – число возможных состояний кода аутентификации;

2. Число правил формирования кода аутентификации (число ключей), при которых для двух произвольных входных последовательностей соответствующие им значения кода аутентификации не изменяются, ограничено. Число таких ключей не может превосходить значения $P_{\text{кол}}|H|$, где $P_{\text{кол}}$ – вероятность коллизии, $|H|$ – число всех ключей.

Вероятность коллизии кодов аутентификации в схеме со строго универсальным хешированием определяется как $P_{\text{кол}} \leq \varepsilon$.

В основе предлагаемой методики статистического исследования коллизионных свойств формируемых элементов $h(x)$ лежит эмпирическая оценка максимумов числа ключей (правил хеширования) при которых:

1. Для произвольных $x_1, x_2 \in A$, $x_1 \neq x_2$ выполняется равенство

$$h(x_1) = h(x_2) ; \quad (1)$$

2. Для произвольных $x_1 \in A$ и $y_1 \in B$ выполняется равенство

$$h(x_1) = y_1 ; \quad (2)$$

3. Для произвольных $x_1, x_2 \in A$, $x_1 \neq x_2$ и $y_1, y_2 \in B$ выполняются равенства

$$h(x_1) = y_1, h(x_2) = y_2 . \quad (3)$$

Оценка по первому критерию соответствует проверке выполнимости условия для универсального класса хеш-функций, оценка по второму и третьему критерию – условий для строго универсального класса хеш-функций.

Введем следующие обозначения:

$$n_1(x_1, x_2) = \left|\{h \in H : h(x_1) = h(x_2)\}\right|,$$
$$x_1, x_2 \in A, \; x_1 \neq x_2 ;$$
$$n_2(x_1, y_1) = \left|\{h \in H : h(x_1) = y_1\}\right|,$$
$$x_1 \in A, \; y_1 \in B ;$$
$$n_3(x_1, x_2, y_1, y_2) = \left|\{h \in H : h(x_1) = y_1, h(x_2) = y_2\}\right|,$$
$$x_1, x_2 \in A, \; x_1 \neq x_2, \; y_1, y_2 \in B .$$

Первый показатель $n_1(x_1, x_2)$ характеризует число правил хеширования, при которых для заданных $x_1, x_2 \in A$, $x_1 \neq x_2$ выполняется равенство (1), т.е. число ключей, при которых существует коллизия (совпадение хеш-кодов) для двух входных последовательностей x_1 и x_2.

Второй показатель $n_2(x_1, y_1)$ характеризует число правил хеширования, при которых для заданных $x_1 \in A$, $y_1 \in B$ выполняется равенство (2), т.е. число ключей, при которых для входной последовательности x_1 значение хеш-кода y_1 не изменяется.

Третий показатель $n_3(x_1, x_2, y_1, y_2)$ характеризует число правил хеширования, при которых для заданных $x_1, x_2 \in A$, $x_1 \neq x_2$, $y_1, y_2 \in B$ выполняется равенство (3), т.е. число ключей, при которых для двух входных последовательностей x_1 и x_2 соответствующие им значения хеш-кодов y_1 и y_2 не изменяются.

Поскольку число ключей, при которых могут выполняться равенства (1), (2) и (3), не должно превосходить соответствующих им значений $P_{\text{кол}} \cdot |H|$, $|H|/|B|$ и $P_{\text{кол}}|H|/|B|$ проведем оценку максимального числа таких ключей для каждого из рассматриваемого набора элементов.

Ограничимся изучением статистических характеристик максимумов этих величин, а затем сравним полученные результаты с числом $P_{\text{кол}} \cdot H$ (для первого критерия), с числом $|H|/|B|$ (для второго критерия) и числом $P_{\text{кол}} \cdot H$ (для третьего критерия).

Таким образом, в качестве статистических показателей оценки коллизионных свойств, по которым будем проводить экспериментальные исследования, предлагается использовать:

– математические ожидания $m(n_1)$, $m(n_2)$ и $m(n_3)$ максимумов числа правил хеширования, при которых выполняются равенства (1), (2) и (3), соответственно;

– дисперсии $D(n_1)$, $D(n_2)$ и $D(n_3)$, характеризующие рассеивание значений числа правил хеширования, при которых выполняются равенства (1), (2) и (3), относительно их математических ожиданий $m(n_1)$, $m(n_2)$ и $m(n_3)$, соответственно.

Оценку коллизионных свойств по приведенным критериям будем производить в среднестатистическом смысле. Другими словами, при постановке эксперимента будем использовать ограниченный набор элементов $x_1, x_2 \in A$, $x_1 \neq x_2$

и соответствующих им хеш-образов $y_1, y_2 \in B$, рассматривая соответствующие результаты как выборку из генеральной совокупности.

Естественной оценкой для математического ожидания m случайной величины X является среднее арифметическое ее наблюдаемых значений X_i (или статистическое среднее) [15]

$$\tilde{m} = \frac{1}{N}\sum_{i=1}^{N} X_i ,$$

где N – количество реализаций случайной величины X.

Оценка дисперсии случайной величины X определяется выражением

$$\tilde{D} = \frac{1}{N-1}\sum_{i=1}^{N} (X_i - \tilde{m})^2 .$$

В силу центральной предельной теоремы теории вероятностей при больших значениях количества реализаций N среднее арифметическое будет иметь распределение, близкое к нормальному закону [15] с математическим ожиданием

$$m[\tilde{m}] \approx \tilde{m}$$

и средним квадратическим отклонением

$$\sigma[\tilde{m}] \approx \frac{\sigma}{\sqrt{N}} ,$$

где σ – среднее квадратическое отклонение оцениваемого параметра.

При этом вероятность того, что оценка $\tilde{m}$ отклонится от своего математического ожидания меньше, чем на ε (доверительная вероятность), равна [15]

$$P(|\tilde{m} - m| < \varepsilon) \approx 2\Phi\left(\frac{\varepsilon}{\sigma[\tilde{m}]}\right), \qquad (4)$$

где $\Phi(x)$ – функция Лапласа, определяется выражением

$$\Phi(x) = \frac{1}{\sqrt{2\pi}}\int_0^x e^{-\frac{t^2}{2}} dt . \qquad (5)$$

Таким образом, при проведении экспериментальных исследований коллизионных свойств будем использовать методы статистической проверки гипотез и математической статистики.

1. Из генеральной совокупности случайной величины X сформируем выборку следующим образом:

– для среднестатистической оценки математического ожидания $m(n_1)$ и дисперсии $D(n_1)$ в качестве случайной величины выступает максимум $n_1(x_1, x_2)$ при которых выполняется равенство $h(x_1) = h(x_2)$, следовательно, выборку объема N: X_1, X_2, ..., X_N сформируем отбором N множеств, в каждом из которых содержится M пар элементов $x_1, x_2 \in A$, $x_1 \neq x_2$ и оценивается $n_1(x_1, x_2)$, т.е. общий объем формируемых пар элементов $x_1, x_2 \in A$, $x_1 \neq x_2$ составит NM;

– для среднестатистической оценки $m(n_2)$ и $D(n_2)$ в качестве случайной величины выступает максимум $n_2(x_1, y_1)$ при которых выполняется равенство $y_1 = h(x_1)$, следовательно, выборку объема N: X_1, X_2, ..., X_N сформируем отбором N множеств, в каждом из которых содержится M пар элементов $x_1 \in A$, $y_1 \in B$ и оценивается $n_2(x_1, y_1)$. Общий объем формируемых пар элементов $x_1 \in A$, $y_1 \in B$ составит NM;

– для среднестатистической оценки $m(n_3)$ и $D(n_3)$ в качестве случайной величины выступает максимум $n_3(x_1, x_2, y_1, y_2)$ при которых выполняются равенства $y_1 = h(x_1)$ и $y_2 = h(x_2)$, следовательно, выборку объема N: X_1, X_2, ..., X_N сформируем отбором N множеств, в каждом из которых содержится M четверок элементов $x_1, x_2 \in A$, $x_1 \neq x_2$, $y_1, y_2 \in B$ и оценивается $n_3(x_1, x_2, y_1, y_2)$, общий объем формируемых четверок составит NM.

2. При экспериментальных исследованиях коллизионных свойств хеширования будем оценивать среднее арифметическое $\tilde{m}(n_i)$ наблюдаемых значений максимумов n_i и дисперсию $\tilde{D}(n_i)$, $i = 1, 2, 3$.

3. Достоверность полученных среднестатистических оценок обоснуем следующим образом. Зафиксируем точность ε и рассчитаем значения функции Лапласа, которые, в соответствии с выражением (4), дадут соответствующие доверительные вероятности:

$$P(|\tilde{m}(n_i) - m(n_i)| < \varepsilon) \approx 2\Phi\left(\frac{\varepsilon}{\sigma[\tilde{m}(n_i)]}\right),$$

$$\sigma[\tilde{m}(n_i)] \approx \frac{\sqrt{\tilde{D}(n_i)}}{\sqrt{N}} .$$

При обратной постановке задачи, т.е. для фиксированной доверительной вероятности $P_д$ при объеме выборки N доверительный интервал определим следующим образом:

$$\tilde{m}(n_i) - t_р \cdot \sigma[\tilde{m}(n_i)] < m(n_i) < \tilde{m}(n_i) + t_р \cdot \sigma[\tilde{m}(n_i)] , \qquad (6)$$

где $t_р$ – корень уравнения $2\Phi(t_р) = P_д$.

Таким образом, предлагаемая методика, используя уменьшенные модели отдельных слоев преобразований, на основе оценки распределения столкновений формируемых образов позволяет экспериментально исследовать коллизионные свойства кодов аутентификации сообщений.

4. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ И ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

С использованием разработанной уменьшенной модели UMAC (mini-UMAC) и методики статистического исследования коллизионных свойств кодов аутентификации сообщений проведем экспериментальную оценку распределения числа столкновений (коллизий) формируемых образов.

Поскольку в рассмотренной выше схеме UMAC на первом слое (при формировании

хеш-кода $Y_{\min i}$) используются семейства универсальных хеширующих функций, подробно исследуемые в работах [1-7], статистические исследования проведем только на втором слое (при формировании псевдослучайной подложки $Pad_{\min i}$) и на заключительном этапе формирования кодов аутентификации (после выполнения суммирования $Tag_{\min i} = Y_{\min i} \oplus Pad_{\min i}$). Именно на этих этапах, по нашему предположению и нарушаются свойства универсальности формируемых кодов аутентификации.

При проведении статистических исследований коллизионных свойств формируемых значений $Pad_{\min i}$ и $Tag_{\min i}$ для каждого эксперимента оценивались математические ожидания $m(n_1)$, $m(n_2)$ и $m(n_3)$, дисперсии $D(n_1)$, $D(n_2)$ и $D(n_3)$, а также для фиксированной точности $\varepsilon = 0{,}1$ рассчитывались соответствующие доверительные вероятности $P(|\tilde{m}(n_i) - m(n_i)| < \varepsilon)$. Исследования проводились над выборкой, объема $N = 100$, для формирования каждого элемента выборки рассчитывался максимум по множеству из $M = 1000$ кортежей элементов. Таким образом, общий объем формируемых наборов составил $NM = 10^5$.

Полученные результаты экспериментальных исследований сведены в табл. 2.

Таблица 2

Результаты экспериментальных исследований коллизионных свойств кодов аутентификации, сформированных с использованием mini-AES и mini-UMAC

	mini-AES, $Pad_{\min i}$	mini-UMAC, $Tag_{\min i}$
$\tilde{m}(n_1)$	–	4,23
$\tilde{D}(n_1)$	–	0,18
$P_д = P(\|\tilde{m}(n_1) - m(n_1)\| < \varepsilon)$	–	0,98
$\tilde{m}(n_2)$	6,68	4,78
$\tilde{D}(n_2)$	0,42	0,42
$P_д = P(\|\tilde{m}(n_2) - m(n_2)\| < \varepsilon)$	0,88	0,88
$\tilde{m}(n_3)$	0,19	5,31
$\tilde{D}(n_3)$	0,15	0,24
$P_д = P(\|\tilde{m}(n_3) - m(n_3)\| < \varepsilon)$	0,99	0,96

При исследовании коллизионных свойств кодов аутентификации, сформированных с использованием мини-версии шифра AES, число ключей, для которых выполняется равенство $h(x_1) = h(x_2)$, при всех испытаниях равнялось нулю, т.е. $n_1(x_1, x_2) = 0$ во всех $N = 100$ опытах. Этот результат объясняется следующим свойством. Шифр AES (как и его мини-версия), реализует биективное отображение множества открытых текстов во множество шифрограмм, т.е. для фиксированного ключа формируемые шифртексты, соответствующие различным открытым текстам, будут различны. Проводимые экспериментальные исследования по первому введенному критерию как раз и состояли в подсчете числа ключей, при которых наблюдается столкновение (коллизия) двух шифр-текстов, соответствующих двум различным открытым текстам, что невозможно по определению биективного шифра. В связи с этим статистические данные по первому критерию для мини-версии шифра AES в таблице 2 не приведены как не информативные.

Анализ приведенных в таблице 2 данных позволяет утверждать об адекватности полученных результатов и соответствии их статистическим свойствам всей генеральной совокупности данных. Для фиксированной точности $\varepsilon = 0{,}1$ получены высокие значения доверительной вероятности, что свидетельствует об обоснованности и достоверности полученных экспериментальных результатов.

Проанализируем полученные результаты статистических исследований коллизионных свойств кодов аутентификации сообщений, сравним полученные результаты среднестатистических оценок математических ожиданий $m(n_1)$, $m(n_2)$ и $m(n_3)$ числа правил хеширования, при которых выполняются равенства (1), (2) и (3), соответственно, с теоретическим оценками: числом $P_{кол} \cdot |H|$ (для первого критерия), с числом $|H|/|B|$ (для второго критерия) и числом $P_{кол} \cdot H$ (для третьего критерия).

Рассмотрим *первый критерий*, по которому оценивается число правил хеширования, при которых существует коллизия (совпадение кодов аутентификации) для двух произвольных входных последовательностей. В соответствии с теоретическими оценками эта величина ограничена сверху числом $P_{кол} \cdot |H|$. Конкретизируем эту (теоретическую) оценку для кодов аутентификации, сформированных с использованием mini-AES и mini-UMAC.

Мощность ключевого множества для mini-AES и mini-UMAC составляет $|H| = 2^{16}$, мощность множества формируемых кодов аутентификации также составляет $|B| = 2^{16}$. Если использовать верхнюю оценку вероятности коллизий как обратную величину мощности формируемых кодов аутентификации $P_{кол} = 2^{-16}$ получим $n_1(x_1, x_2) \le P_{кол} \cdot |H| = 1$. Для мини-версии шифра AES это условие выполняется (обосновывается биективностью шифрующего преобразования), однако коллизионные свойства mini-UMAC существенно уступают этой верхней теоретической оценке. Фактически, число коллизий выше теоретической границы более чем в четыре раза и это положение подтверждено с высокой доверительной вероятностью $P_д = P(|\tilde{m}(n_1) - m(n_1)| < 0{,}1) > 0{,}98$.

Рассмотрим *второй критерий*, по которому оценивается число правил хеширования, при которых для произвольной входной последовательности значение кода аутентификации не изменяется. В соответствии с теоретическими оценками эта величина для кодов аутентификации, сформированных с использованием mini-AES и mini-UMAC, ограничена сверху числом $|H|/|B|=1$. Полученные экспериментальные результаты свидетельствуют, что коллизионные свойства кодов аутентификации, сформированных с использованием mini-AES и mini-UMAC, не удовлетворяют второму критерию, число ключей, при которых для произвольной входной последовательности значение кода аутентификации не изменяется в несколько раз превышает теоретическую оценку для универсального хеширования.

В соответствии с *третьим критерием* оценивается число правил хеширования, при которых для двух произвольных входных последовательностей соответствующие им значения кода аутентификации не изменяются. Теоретическая оценка этой величины для универсального хеширования ограничена сверху числом $P_{\text{кол}}|H|$, что при использовании верхней оценки вероятности коллизий $P_{\text{кол}}=2^{-16}$ дает $n_3(x_1,x_2,y_1,y_2)\le P_{\text{кол}}\cdot|H|=1$. Значения, приведенные в таблице 2, свидетельствуют о том, что коллизионные свойства кодов аутентификации, сформированных с использованием mini-AES, удовлетворяют третьему критерию. В тоже время число ключей mini-UMAC, при которых для двух произвольных входных последовательностей соответствующие им значения кода аутентификации не изменяются, более чем в пять раз выше верхней теоретической оценки.

ВЫВОДЫ

Таким образом, из полученных результатов статистических исследований коллизионных свойств кодов аутентификации сообщений, сформированных с использованием mini-AES и mini-UMAC, можно сделать следующие важные в прикладном отношении выводы:

– криптографический слой формирования кодов аутентификации сообщений (mini-AES) удовлетворяет свойствам универсального хеширования, вероятность коллизии формируемых хеш-образов не превосходит наперед заданной величины (первый критерий).

Это объясняется, прежде всего, тем, что шифрование неповторяющегося (уникального) для всех информационных сообщений значения *Nonce* приводит к формированию множества неповторяющихся (уникальных) для всех информационных сообщений псевдослучайных подложек *Pad*.

Другими словами, формирование псевдослучайных подложек *Pad* осуществляется в результате биективного отображения множества неповторяющихся (уникальных) для всех информационных сообщений значений *Nonce*, в результате чего коллизии (столкновения) подложек *Pad* отсутствуют по определению. В тоже время, данный слой преобразований не удовлетворяет свойствам строго универсального хеширования (не выполняется второй критерий) (см. табл. 2). Кроме того, обеспечение свойств универсального хеширования на этом слое предполагает формирование и передачу неповторяющегося для каждого сообщения значения *Nonce*, что требует дополнительных временных и программно-аппаратных затрат;

– результат формирования кодов аутентификации сообщений по схеме mini-UMAC не удовлетворяет свойствам как универсального хеширования, так и, тем более, свойствами строго универсального хеширования. Это объясняется тем, что схема с простым суммированием по модулю два (XOR) двух результатов универсального хеширования не всегда сохраняет свойства универсального хеширования.

Поясним последний вывод на примере. Пусть первый и второй слой схемы формирования кодов аутентификации обладают свойствами универсального хеширования. Условно обозначим процесс такого хеширования в виде таблиц 3 и 4, где столбцами обозначены информационные сообщения $M_1, M_2, \ldots, M_n$, а строками – правила хеширования (h_i и g_j, соответственно), заданные (параметризированные) соответствующими секретными ключами. В ячейках таблиц содержатся результаты хеширования, т.е. искомые хеш-коды.

Таблица 3

	M_1	M_2	M_3	...	M_n
h_1	$h_1(M_1)$	$h_1(M_2)$	$h_1(M_3)$	...	$h_1(M_n)$
h_2	$h_2(M_1)$	$h_2(M_2)$	$h_2(M_3)$	...	$h_2(M_n)$
h_3	$h_3(M_1)$	$h_3(M_2)$	$h_3(M_3)$	...	$h_3(M_n)$
...	...	...	...	...	...
h_k	$h_k(M_1)$	$h_k(M_2)$	$h_k(M_3)$	...	$h_k(M_n)$

Таблица 4

	$M1$	M_2	M_3	...	M_n
g_1	$g_1(M_1)$	$g_1(M_2)$	$g_1(M_3)$	...	$g_1(M_n)$
g_2	$g_2(M_1)$	$g_2(M_2)$	$g_2(M_3)$	...	$g_2(M_n)$
g_3	$g_3(M_1)$	$g_3(M_2)$	$g_3(M_3)$	...	$g_3(M_n)$
	...	...	...	...	...
g_k	$g_k(M_1)$	$g_k(M_2)$	$g_k(M_3)$	...	$g_k(M_n)$

Если каждая пара правил хеширования h_i и g_i задается одним секретным ключом K_i, тогда результирующая схема с простым суммированием по модулю два (XOR) двух результатов универсального хеширования информационных сообщений может быть представлена табл. 5.

Таким образом, общее число правил хеширования не изменилось (по сравнению с числом правил хеширования для функций $h(x)$ и $g(x)$, соответственно), оно определяется мощностью множества используемых секретных ключевых данных. Каждый секретный ключ K_i задает

Таблица 5

		M_1	M_2	M_3	...	M_n
K_1	h_1, g_1	$h_1(M_1)\oplus g_1(M_1)$	$h_1(M_2)\oplus g_1(M_2)$	$h_1(M_3)\oplus g_1(M_3)$	...	$h_1(M_n)\oplus g_1(M_n)$
K_2	h_2, g_2	$h_2(M_1)\oplus g_2(M_1)$	$h_2(M_2)\oplus g_2(M_2)$	$h_2(M_3)\oplus g_2(M_3)$	...	$h_2(M_n)\oplus g_2(M_n)$
K_3	h_3, g_3	$h_3(M_1)\oplus g_3(M_1)$	$h_3(M_2)\oplus g_3(M_2)$	$h_3(M_3)\oplus g_3(M_3)$	...	$h_3(M_n)\oplus g_3(M_n)$
...	...	...	...	...	...	...
K_k	h_k, g_k	$h_k(M_1)\oplus g_k(M_1)$	$h_k(M_2)\oplus g_k(M_2)$	$h_k(M_3)\oplus g_k(M_3)$	...	$h_k(M_n)\oplus g_k(M_n)$

(параметризирует) два правила h_i и g_i, которые применяются к каждому информационному сообщению, подлежащему хешированию. Результат преобразования представлен в соответствующих ячейках таблицы 5 как результат суммирования по модулю 2 (XOR) значений $h_i(M_j)$ и $g_i(M_j)$.

Очевидно, что коллизия хеш-кодов будет наблюдаться для всех сообщений M_i и M_j, для которых выполняется равенство:

$$h_w(M_i)\oplus g_w(M_i) = h_w(M_j)\oplus g_w(M_j). \qquad (7)$$

Даже если функции h_w и g_w для сообщений M_i и M_j не вызывают коллизию, т.е., если

$$h_w(M_i) \neq h_w(M_j)$$

и

$$g_w(M_i) \neq g_w(M_j)$$

равенство (7) все равно может выполняться, и число правил (и число соответствующих ключей), вызывающих коллизию в результирующей схеме, возрастет. Это событие будет достоверным (произойдет наверняка), например, в случае, если

$$h_w(M_i) = h_w(M_j)$$

и

$$g_w(M_i) = g_w(M_j)$$

Таким образом, схема с простым суммированием по модулю два (XOR) двух результатов универсального хеширования в общем случае не обеспечивает сохранение свойств универсального хеширования. Коллизионные свойства кодов аутентификации сообщений снижаются и, как показывает анализ таблицы 2, не удовлетворяют поставленным требованиям.

Таким образом, нарушение коллизионных свойств универсального хеширования в схеме mini-UMAC (после применения криптографического слоя преобразования) следует считать экспериментально доказанным.

Перспективным направлением дальнейших исследований является разработка методов построения криптографически стойких схем формирования кодов аутентификации с обеспечением высоких коллизионных свойств универсального хеширования. Одним из перспективных направлений в этом смысле является использование модулярных преобразований.

Литература

[1] *Black J.* "UMAC: Fast and provably secure message authentication", Advances in Cryptology / J. Black, S. Halevi H., Krawczyk, T. Krovetz, P. Rogaway. – CRYPTO '99, LNCS vol. 1666, PP. 216-233, Springer-Verlag, 1999.

[2] *T. Krovetz, P. Rogaway.* Fast universal hashing with small keys and no preprocessing", work in progress, 2000. – URL: http://www.cs.ucdavis.edu/~rogaway/umac

[3] *T. Krovetz, J. Black, S. Halevi, A. Hevia, H. Krawczyk, P. Rogaway.* UMAC -Message authentication code using universal hashing. IETF Internet Draft, draft-krovetz-umac-00.txt. – URL: www.cs.ucdavis.edu/~rogaway/umac, 2000.

[4] *Krovetz T.* UMAC -Message authentication code using universal hashing. IETF Internet Draft, draft-krovetz-umac-02.txt. – URL: www.cs.ucdavis.edu/~rogaway/umac, 2004.

[5] Final report of European project number IST-1999-12324, named New European Schemes for Signatures, Integrity, and Encryption, April 19, 2004 – Version 0.15 (beta), Springer-Verlag.

[6] *Krovetz T.* UMAC - Message authentication code using universal hashing, 2006. – URL: http://www.cs.ucdavis.edu/~rogaway/umac

[7] *Krovetz T.* Software-Optimized Universal Hashing and Message Authentication. Dissertation submitted in partial satisfaction of the requirements for the degree of doctor of philosophy. University Of California Davis. September 2000. – 269 p.

[8] *Carter J. L.* Universal classes of hash functions / J.L. Carter, M.N. Wegman // Computer and System Scince – 1979 – №18 – P. 143–154.

[9] *Wegman M. N.* New hash functions and their use in authentication and set equality / M. N.Wegman, J. L. Carter / Computer and System Scince – 1981 – № 22 – P. 265-279.

[10] A Description of Baby Rijndael // ISU CprE/Math 533; NTU ST765-U. – 2003.

[11] Raphael Chung-Wei Phan, "Mini Advanced Encryption Standard (Mini-AES): A testbed for Cryptanalysis Students", Cryptologia, XXVI (4), October 2002. – PP. 283-306.

[12] *Долгов В.И.* Исследование дифференциальных свойств мини-шифров Baby-ADE и Baby-AES / В.И. Долгов, А.А. Кузнецов, Р.В. Сергиенко, О.И. Олешко // Прикладная радиоэлектроника. – Х.: ХНУРЭ, 2009. – Т. 8, № 3. – С. 252–257.

[13] *Долгов В.И.* Подход к криптоанализу современных шифров / В.И. Долгов, И.В. Лисицкая, Р.В. Олейников. // Материалы второй международной конференции «Современные информационные системы. Проблемы и тенденции развития», Харьков-Туапсе, Украина, 2–5 октября. – 2007. – С. 435–436.

[14] *Сорока Л.С.* Исследование дифференциальных свойств блочно-симметричных шифров. / Сорока Л.С., Кузнецов А.А., Московченко И.В., Исаев С.А. // Системи обробки інформації. – Харків: ХУ ПС. – 2010 – Вип. 6(87). – С. 286–294.

[15] *Вентцель Е.С.* Теория вероятностей / Е.С. Вентцель. – М.: Государственное издательство физико-математической литературы, 1958 – 564 с.

Поступила в редколлегию 9.04.2012

Кузнецов Александр Александрович, доктор технических наук, профессор, профессор кафедры БИТ ХНУРЭ. *Область научных интересов:* теория помехоустойчивого кодирования, криптография и аутентификация.

Король Ольга Григорьевна, преподаватель кафедры информационных систем ХНЕУ. *Область научных интересов:* теория аутентификации, методы и вычислительные алгоритмы хеширования информации.

Евсеев Сергей Петрович, кандидат технических наук, с.н.с., доцент кафедры информационных систем ХНЕУ. *Область научных интересов:* теория кодирования, криптография и аутентификация.

УДК 681.3.06

Дослідження колізійних властивостей кодів автентифікації повідомлень UMAC / О.О. Кузнецов, О.Г. Король, С.П. Євсєев // Прикладна радіоелектроніка: наук.-техн. журнал. – 2012. – Том 11. № 2. – С. 171–183.

Розглядається алгоритм формування кодів автентифікації повідомлень UMAC, в основі якого лежить використання універсальних гешуючіх функцій. Пропонується зменшена модель UMAC (mini-umac) і методика статистичного дослідження колізійних властивостей формованих кодів автентифікації повідомлень. З використанням зменшеної моделі UMAC досліджуються колізійні властивості кодів автентифікації, показано, що застосування криптографічного перетворення (з використанням алгоритму AES) на завершальному етапі UMAC приводить до порушення властивостей універсального гешування.

Ключові слова: міні-UMAC, автентифікація, універсальна функція, коди автентичності, алгоритм AES.

Таб. 5. Бібліогр.: 15 найм.

UDC 681.3.06

Studying collision characteristics of authentication codes of messages UMAC / A.A. Kuznetsov, O.G. Korol, S.P. Evseev // Applied Radio Electronics: Sci. Journ. – 2012. Vol. 11. № 2. – P. 171–183.

The paper considers the algorithm of forming authentication codes of messages UMAC which is based on use of universal hashing functions. A reduced model UMAC (mini-UMAC) and methods of statistical research of collision characteristics of formed authentication message codes are suggested. The collision characteristics of authentication codes are researched with the help of using the reduced model UMAC. It is shown that using cryptographic transformation (with the application of the AES algorithm) at the final UMAC stage results in violation of universal hashing properties.

Keywords: mini-UMAC, authentication, universal function, authentication codes, AES algorithm.

Tab. 5. Ref.: 15 items.