УДК 621.391

О.Г. Король

Харьковский национальный экономический университет имени С. Кузнеца, Харьков

ОЦЕНКА ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ НЕКОТОРЫХ ФУНКЦИЙ ХЕШИРОВАНИЯ

Проводится анализ вычислительной сложности некоторых алгоритмов хеширования, применяемых в коммуникационных системах на основе оценки временных и скоростных показателей пропускной способности процессора, сравнительная оценка вычислительной сложности усовершенствованного алгоритма UMAC с использованием в качестве псевдослучайной подложки алгоритмов модулярного преобразования MASH-1 и MASH-2, и статистической безопасности на основе пакета NIST STS.

Ключевые слова: *модулярные преобразования, ключевое хеширование, число тактов процессора, пропускная способность алгоритма.*

Вступление

Проведенные исследования показали, что применение многослойных схем ключевого хеширования позволяет строить эффективные механизмы контроля целостности и аутентичности информации в телекоммуникационных системах и сетях. Однако известные многослойные конструкции (на примере алгоритма UMAC) наряду с высокими показателями быстродействия и криптографической стойкости за счет применения криптографического слоя преобразования (с использованием блочного симметричного шифра) теряют свойства универсального хеширования, что приводит к ухудшению коллизионных свойств формируемых кодов аутентификации сообщений. Предлагаемый в работах [1, 4] метод универсального хеширования с использованием модулярных преобразований алгоритмов MASH-1 и MASH-2 позволяет реализовать формирование аутентификаторов (хеш-кодов) с обеспечением требуемых показателей стойкости.

Целью работы является анализ вычислительной сложности некоторых алгоритмов хеширования, применяемых в коммуникационных системах на основе оценки временных и скоростных показателей пропускной способности процессора, сравнительная оценка вычислительной сложности усовершенствованного алгоритма UMAC с использованием в качестве псевдослучайной подложки алгоритмов модулярного преобразования MASH-1 и MASH-2 и статистической безопасности на основе пакета NIST STS.

Основная часть

Анализ вычислительной сложности некоторых алгоритмов хеширования. Для сравнения схем ключевого хеширования по показателям стойкости и быстродействия принято использовать единицу измерения cpb, где cpb (cycles per byte) – определяет число тактов процессора, необходимых для обработки 1 байта входной информации. Сложность алгоритма рассчитывается по формуле

$$\mathrm{Per} = \mathrm{Utl} * \mathrm{CPU\_clock} / \mathrm{Rate},$$

где Utl – утилизация ядра процессора, усреднённая утилизация процессора за интервал времени – на каждом отрезке, на котором не выполняется Idle Thread, процессор считается занятым какой-то реальной нагрузкой. Этот счётчик – сумма показателей утилизации ЦПУ пользователем, системой и во время простоя (Idle + User + System utilization, названия могут отличаться на разных платформах). Учитывая, что на большинстве платформ есть отдельный счётчик ЦПУ простоя, рекомендуется использовать следующую формулу для расчета потребления ЦПУ

$$\text{CPU Consumption} = 100 - \text{Idle CPU (\%)}.$$

Rate – пропускная возможность алгоритма (байт/сек). Для оценки пропускной способности, измеряемой в машинных циклах за байт, для обработки сообщений различной длины байт используем циклы за байт, поскольку это дает возможность сравнения эффективности между процессорами работающих на различных скоростях.

Чтобы преобразовать в байты в секунду, необходимо разделить циклы за секунду процессора (Гц) на переданные циклы за байт. Например, процессор с тактовой частотой в 1 ГГц со скоростью 2,0 цикла за байт выполняет 1e9/2,0 = 0.5e9 байт в секунду (500 МБ/сек).

Пример значений по показателям стойкости и быстродействия одного из алгоритмов-конкурсантов конкурса NIST на национальный стандарт алгоритма хеширования SHA-3 приведен в табл. 1. Биты безопасности приведены по категориям стойкости БСШ: 80, 112, 128, 192, 256 [6].

Анализ таблицы показывает, что увеличение показателей стойкости алгоритма приводит и к увеличению его сложности.

Таблица 1

Показатели стойкости и быстродействия алгоритма хеширования Blake

Алгоритм хеширования	Уровень безопасности Sec [біт]	Утилизация ядра, Utl (%)	Пропускная способность Rate (байт/сек)	Сложность алгоритма, Per (cpb)
Hash (Blake-224)	80, 112, 128, 192	56	19286741	61,5
Hash (Blake-256)	80, 112, 128, 192, 256	56	19192519	62,0
Hash (Blake-384)	80, 112, 128, 192, 256	56	15610497	76,0

Сравнительная оценка вычислительной сложности усовершенствованного алгоритма UMAC. В разработанном усовершенствованном методе формирования кодов контроля целостности и аутентичности данных первые слои преобразования предлагается реализовать традиционными для алгоритма UMAC высокоскоростными, но криптографически слабыми схемами универсального хеширования, последний слой предлагается реализовать с использованием разработанной безопасной (криптографически сильной) схемы строго универсального хеширования на основе модулярных преобразований.

Формально предлагаемая схема каскадного формирования кодов контроля целостности и аутентичности данных представлена на рис. 1.

Рис. 1. Усовершенствованная схема каскадного формирования кодов контроля целостности и аутентичности данных с использованием модулярных преобразований

Для сравнения с другими схемами ключевого хеширования по показателям стойкости и быстродействия можно принять следующие допущения.

Пусть одна операция умножения над числами, порядка 2^m требует $\lceil m/L \rceil$ операций поразрядного сложения по модулю два (XOR), где L – разрядность процессора используемой вычислительной системы, $\lceil x \rceil$ – округленное до б*о*льшего целого число x. Подобное допущение наиболее часто применяется при оценке сложности реализации криптоалгоритмов [2, 3]. В этом случае оценка $\lceil m/L \rceil$ дает приблизительное число циклов L-разрядного процессора, необходимых для реализации одного умножения над числами, битовая длина которых не превосходит m. В то же время, при хешировании с использованием модулярных преобразований обрабатывается сразу $m/8$ байт информационных данных.

В табл. 2, 3 приведены результаты сравнительных исследований практической реализации алгоритмов в зависимости от количества логических операций и структуры схем хеширования.

В табл. 4 приведены результаты сравнительных исследований быстродействия схем ключевого хеширования для фиксированных показателей безопасности. Показатель быстродействия выражен в количестве S циклов 32-разрядного процессора, не-

обходимых для формирования одного байта выходных данных. Показатель безопасности фиксировался через длину секретного ключа, который необходимо взломать злоумышленнику.

Для схем на модулярной арифметике приведена эквивалентная длина ключа блочного симметричного криптоалгоритма.

Приведенные данные в табл. 4 свидетельствуют, что использование модулярных преобразований для решения задач ключевого хеширования существенно повышает сложность вычислений, быстродействие алгоритмов снижается на 1-2 порядка. В то же время, разработанные схемы ключевого хеширования обладают доказуемо стойким уровнем безопасности (задача нахождения ключа хеширования или прообраза сводится к решению известной теоретико-сложностной задаче). Кроме того, в работах [1, 4] доказано, что такие схемы аутентификации удовлетворяют свойствам универсального хеширования, чем обеспечиваются высокие коллизионные характеристики формируемых MAC.

Оценим сложность реализации усовершенствованного алгоритма UMAC при увеличении объема входных данных.

Таблица 2

Количество логических операций при практической реализации алгоритмов хеширования

Алгоритм	Логические операции									
	AND	OR	XOR	ROTR	SHR	+	ROLs	NOT	MOD	MUL
MD-5	-	-	-	-	-	960	256	-	-	-
RIPEMD-128	-	-	-	-	-	396	128	-	-	-
RIMEMD-160	-	-	-	-	-	650	320	-	-	-
SHA-1	400	240	320	-	-	320	160	-	-	-
SHA-256	320	-	448	384	-	448	-	64	-	-
SHA-384	400	-	560	480	-	560	-	80	-	-
SHA-512	400	-	560	480	-	560	-	80	-	-
MASH1	-	6	4	-	6	1	1	-	6	5
MASH2	-	6	4	-	6	1	1	-	6	260

 – конкатенация, + – сложение,
and – побитовое «И», *or* – побитовое «ИЛИ»,
xor – исключающее «ИЛИ», shr (Shift Right) – логический сдвиг вправо,
rotr (Rotate Right) – циклический сдвиг вправо, ROLs – циклический сдвиг влево на s позиций

Таблица 3

Количество шагов и циклов в схемах хеширования

Алгоритм	Общее кол-во шагов,	количество циклов r	Кол-во шагов в цикле s	Константы отличные для каждого
MD-	64	4	16	шаг
RIPEMD-128	64	4	16	Цикл
RIMEMD-160	80	5	16	Цикл
SHA-1	80	4	20	Цикл
SHA-2 - 256	64	1	64	Шаг
SHA-2 - 384	80	1	80	Шаг
SHA-2 - 512	80	1	80	Шаг

Таблица 4

Оценка сложности алгоритмов хеширования в количестве S циклов 32-разрядного процессора на один байт обрабатываемых данных

Функция хеширования	Уровень стойкости (длина ключа)	Количество циклов S
SHA-2 (512)	512	80
SHA-2 (256)	256	64
SHA-1	160	80
RIPEMD-160	160	160
MD5	128	64
Хеширование на модулярной арифметике	80	512
	128	1536
	256	7680

В основе разработанной схемы формирования MAC с использованием модулярных преобразований лежит использование:

– на первых слоях – высокоскоростных методов универсального хеширования (NH-хеширование, полиномиальное хеширование, хеширование Картера-Вергмана);

– на последнем слое – безопасного строго универсального хеширования на основе модулярных преобразований. Применение многослойной конструкции позволяет также существенно снизить вычислительные затраты на формирования кодов контроля целостности и аутентичности больших массивов данных.

Поясним последний тезис следующими рассуждениями. Пусть первые слои универсального хеширования (как и в методе-прототипе UMAC) реализуются с использованием высокоскоростных (но криптографически слабых) схем Картера-Вергмана, полиномиальных конструкций и пр. (см. рис. 1). Положим сложность таких преобразований равную сложности схемы UMAC, т.е. порядка 6 циклов на один байт информационных данных. На самом деле эта оценка сильно завышена, поскольку наиболее затратным в схеме UMAC является последний криптографических слой, с использованием алгоритма шифрования AES. Другими словами, оценка в 6 циклов на один байт обрабатываемых данных является оценкой в худшем случае, т.е. оценкой «сверху» [1, 4].

Положим также, что на последнем, криптографическом этапе вместо алгоритма AES используется предложенная схема доказуемо стойкого универсального хеширования на основе модулярных преобразований (см. модель на рис. 1). Для оценки сложности этого, завершающего этапа формирования MAC, используем данные табл. 4.

Тогда результирующая сложность как количество циклов процессора на один обрабатываемый байт данных есть усредненная оценка по всем слоям преобразования в предлагаемой каскадной конструкции вычисления кодов контроля целостности и аутентичности данных.

Поскольку основная часть обрабатываемых данных поступает только на первые слои преобразования (см. модель на рис. 1), а последний, криптографический слой с модулярными преобразованиями используется лишь единожды для обработки результата хеширования предыдущими слоями схемы, результирующая оценка сложности для больших объемов, обрабатываемых данных будет стремиться к оценке сложности схемы UMAC.

Для подтверждения приведенных рассуждений в табл. 5 приведена приблизительная оценка сложности формирования кодов контроля целостности и аутентичности, данных предложенной схемой с использованием модулярных преобразований.

Таблица 5

Оценка сложности формирования MAC предложенной схемой в количестве циклов S циклов 32-разрядного процессора на один байт обрабатываемых данных

Уровень стойкости (эквивалентная длина ключа, бит)	Длина входных данных, байт									
	128	256	512	1024	2048	4096	8192	16384	32768	65536
80	518	262	134	70	38	22	14	10	8	7
128	–	–	1158	582	294	150	78	42	24	15
256	–	–	–	–	7206	3606	1806	906	456	231

Данные, приведенные в табл. 5, получены расчетным путем посредством усреднения верхней оценки сложности универсального хеширования на первых слоях преобразований (6 циклов на один байт) и оценки сложности модулярных преобразований (с использованием цикловых функций из табл. 4. Прочерками в табл. 5 проставлены места, в которых хеширование на модулярных преобразованиях (криптографический слой) не может быть выполнено. Так, например, модулярное преобразование для уровня стойкости с эквивалентной длиной ключа симметричного шифра в 128 бит необходимо реализовать для длины модуля не менее 3072 бит (см. табл. 4), что при входных данных их 256 байт (2048 бит) данных невозможно. Анализ данных табл. 5 подтверждает приведенные выше рассуждения о снижении удельной сложности преобразования (количестве циклов процессора на один байт входных данных) при увеличении длины обрабатываемых информационных данных. Практически это означает, что с ростом длины блоков данных предлагаемая схема формирования кодов контроля целостности и аутентичности по вычислительной сложности становиться эквивалентной применяемым на сегодняшний день в протоколах сетевой безопасности (в том числе в протоколах IPsec) алгоритмам MD-5 и SHA-1, а также алгоритмам SHA-2, CBC MAC-Rijndael и др.

В табл. 6 приведено сравнение вычислительной сложности некоторых функций хеширования. Данные по быстродействию для предлагаемой схемы MAC с модулярными преобразованиями приведены для минимального уровня стойкости (мощность множества ключевых данных блочного симметричного шифра равна 280) и достаточного уровня стойкости (для модулярных преобразований эквивалентная длина ключа блочного симметричного шифра равна 128 битам). Длина формируемого при этом MAC равна 80 и 128 битам, соответственно.

Таблица 6

Оценка сложности формирования MAC различными схемами

Алгоритм	Длина входных данных, байт					
	2048	4096	8192	16384	32768	65536
HMAC-MD5 (128 бит)	9	9	9	9	9	9
HMAC-RIPE-MD (160 бит)	27	27	27	27	27	27
HMAC-SHA-1 (160 бит)	25	25	25	25	25	25
HMAC-SHA-2 (512бит)	84	84	84	84	84	84
CBC MAC-Rijndael (128 бит)	26	26	26	26	26	26
CBC MAC-DES (64 бита)	62	62	62	62	62	62
Усовершенствованная схема UMAC с модулярными преобразованиями (80 бит)	38	22	14	10	8	7
Усовершенствованная схема UMAC с модулярными преобразованиями (128 бит)	294	150	78	42	24	15

Для всех функций, приведенных в табл. 6 (кроме предложенных, с использованием модулярных преобразований) удельная сложность формирования кодов контроля целостности и аутентичности данных не зависит от объема обрабатываемых данных (при заполнении табл. 6 использованы данные из отчета конкурса NESSIE [2]). Для усовершенствованного метода UMAC с использованием модулярных преобразований удельная сложность с ростом длины обрабатываемых данных снижается. Так для высокого уровня стойкости (эквивалентная длина ключа блочного симметричного шифра равна 128 битам) уже для блоков данных из 32768 байт сопоставима с известными и применяемыми в протоколах сетевой безопасности алгоритмами формирования MAC. Для минимального уровня стойкости (мощность множества ключевых данных блочного симметричного шифра равна 280) предлагаемая схема каскадного формирования кодов контроля целостности и аутентичности данных с использованием модулярных преобразований уже для пакетов данных из 2048 байт практически не уступает по быстродействию применяемым на сегодняшний день алгоритмам формирования MAC в протоколах сетевой безопасности, в том числе в протоколах IPsec.

Таким образом, полученные результаты исследований показывают, что разработанная схема формирования кодов контроля целостности и аутентичности данных с использованием модулярных преобразований позволяет обеспечить высокие коллизионные свойства безопасного хеширования. Кроме того, за счет многослойной конструкции вычисления хеш-кода удается существенно сократить вычислительную сложность хеширования и повысить, таким образом, скорость обработки информационных сообщений.

Полученные результаты теоретических исследований позволяют обосновать практические рекомендации по использованию разработанных моделей и методов каскадного формирования кодов контроля целостности, и аутентичности данных для повышения безопасности телекоммуникационных систем и сетей.

Для обеспечения целостности и аутентичности данных в телекоммуникационных сетях используются коды обнаружения манипуляций (MDC), коды аутентификации сообщений (MAC). Так, например, в протоколах сетевой безопасности IPSec для формирования кодов контроля целостности и аутентичности ICV предусмотрены обязательные алгоритмы (для обеспечения совместимости программных продуктов различных производителей): HMAC-MD5, HMAC-SHA-1, а также другие (дополнительные) алгоритмы, например, DES-MAC. Указанные механизмы применяются по умолчанию в целях обеспечения целостности и аутентичности пакетов данных во всех реализациях сетей IPv6.

Оценка статистической безопасности схем хеширования на основе пакета NIST STS

Для проверки устойчивости алгоритмов-претендентов хеширования используем набор тестов NIST STS по определенной методике исследования статистических свойств хэш-функций [5].

Для проведения тестирования были взяты следующие параметры:

– длина тестируемой последовательности $n = 10^6$ бит;

– количество тестируемых последовательностей $m = 100$;

– уровень значимости $\alpha = 0{,}01$.

Таким образом, объем тестируемой выборки составляет:

– $N = 10^6 \times 100 = 10^8$ бит;

– количество (q) для разных длин $q = 189$, таким образом, статистический портрет генератора составляет 18900 значений вероятности P.

Результаты тестирования алгоритмов хеширования сведены в табл. 7.

Полученные результаты подтверждают теоретические исследования стойкости разработанного каскадного метода хеширования UMAC с использованием в последнем слое в качестве псевдоподложки алгоритмы модулярных преобразований MASH-1 и MASH-2.

Таблица 7

Результаты тестирования алгоритмов хеширования

Генератор	Количество тестов, в которых тестирование прошло больше 99% последовательностей	Количество тестов, в которых тестирование прошло больше 96% последовательностей
BBS	134 (71%)	189 (100%)
FIPS 197	126 (67%)	189 (100%)
Blake	130 (69%)	189 (100%)
CubeHash	137 (73%)	189 (100%)
ECHO	139 (74%)	189 (100%)
Grostl	140 (75%)	189 (100%)
Keccak	134 (71%)	187 (98,94%)
MASH-1	101 (53%)	47 (24%)
MASH-2	126 (67%)	189 (100%)
MASH(EC)	141 (74%)	189 (100%)
UMAC 32	167 (88%)	189 (100%)
HMAC-SHA-256	134 (71%)	187 (98%)
EMAC	138 (73%)	189 (100%)
RIPEMD-160	129 (68%)	189 (100%)
UMAC+MASH-2	173 (91%)	189 (100%)

Выводы

На основании данных, приведенных в табл. 6, можно утверждать, что разработанная каскадная схема формирования кодов контроля целостности и аутентичности данных с использованием модулярных преобразований не уступает по быстродействию используемым в протоколах IPSec механизмам, а при увеличении входной последовательности данных дает существенный выигрыш в быстродействии формирования хеш-кода.

В то же время, предлагаемая схема обеспечивает доказуемо стойкий уровень безопасности и коллизионные свойства на уровне строго универсального хеширования.

Поскольку спецификациями протоколов AH и ESP IPSec предусмотрено использование новых, более эффективных алгоритмов формирования ICV, для защиты пакетов данных в коммуникационных сетях предлагается использование разработанных моделей и методов каскадного формирования кодов контроля целостности, и аутентичности данных на основе модулярных преобразований.

Список литературы

1. Кузнецов О.О. Захист інформації в інформаційних системах / О.О. Кузнецов, С.П. Євсеєв, О.Г. Король. – Х.: Вид. ХНЕУ, 2011. – 504 с.

2. Final report of European project number IST-1999-12324, named New European Schemes for Signatures, Integrity, and Encryption, April 19, 2004 – Springer-Verlag.

3. Столлингс В. Криптография и защита сетей: принципы и практика, 2-е изд.: пер. с англ. / В. Столлингс. – М.: издательский дом «Вильям», 2001. – 672 с.

4. Король О.Г. Исследование методов обеспечения аутентичности и целостности данных на основе односторонних хеш-функций / О.Г. Король, С.П. Евсеев // Захист інформації. Спецвипуск (40). – 2008. – С. 50-55.

5. Методика статистического тестирования NIST STS и математические доказательства тестов. – Х.: Інститут информационных технологий, 2004. – 62 с.

6. Status Report on the First Round of the SHA-3 Cryptographic Hash Algorithm Competitionhttp Andrew Regenscheid, Ray Perlner, Shu-jen Chang, John Kelsey, Mridul Nandi, Souradyuti Paul. [Электронный ресурс] – Режим доступа к ресурсу: www.nist.gov/index.html.

Поступила в редколлегию 4.02.2015

Рецензент: д-р техн. наук, проф. В.А. Хорошко, Национальный авиационный университет, Киев.

ОЦІНКА ОБЧИСЛЮВАЛЬНОЇ СКЛАДНОСТІ ДЕЯКИХ ФУНКЦІЙ ГЕШУВАННЯ

О.Г. Король

Проводиться аналіз обчислювальної складності деяких алгоритмів гешування, які використовуються в комунікаційних системах на основі оцінки часових і швидкісних показників пропускної здатності процесора, порівняльна оцінка обчислювальної складності вдосконаленого алгоритму UMAC з використанням як псевдовипадкової підкладки алгоритмів модулярного перетворення MASH-1 і MASH-2, і статистичної безпеки на основі пакету NIST STS.

Ключові слова: *модулярні перетворення, ключове гешування, кількість тактів процесора, пропускна здатність алгоритму.*

EVALUATION OF THE COMPUTATIONAL COMPLEXITY OF SOME HASH FUNCTIONS

O.G. Korol

The analysis of the computational complexity of some hashing algorithms used in communication systems based on an assessment of time and speed performance processor bandwidth, a comparative evaluation of the computational complexity of the improved algorithm, UMAC, using as substrate pseudorandom algorithms modular transformations MASH-1 and MASH-2, and statistical security based package NIST STS.

Keywords: *modular transformations, key hashing, the number of processor cycles, bandwidth algorithm.*