УДК 004.021+681.3.05

А.В. Антонов[1], В.В. Слободянюк[2]

[1] *Харьковский университет Воздушных Сил им. И. Кожедуба, Харьков*
[2] *Воинская часть А1231, Винница*

# АНАЛИЗ СТАТИСТИЧЕСКОЙ БЕЗОПАСНОСТИ МЕТОДА ПОСТРОЕНИЯ ХЕШ-ФУНКЦИИ НА ОСНОВЕ ХАОТИЧЕСКИХ ОТОБРАЖЕНИЙ С ПЕРЕМЕННЫМИ ПАРАМЕТРАМИ И ПАРАЛЛЕЛЬНОЙ ОРГАНИЗАЦИЕЙ ВЫЧИСЛЕНИЙ (С ПОМОЩЬЮ ПАКЕТА ТЕСТИРОВАНИЯ NIST STS)

*Пакетом статистического тестирования NIST STS исследуется базовый и усовершенствованный метод построения хеш-функций на основе хаотических отображений с переменными параметрами и параллельной организацией вычислений. Показано, что предпринятые по усовершенствованию метода меры позволяют достичь высоких характеристик его статистической безопасности. По результатам исследований сформулированы рекомендации по применению метода и выбору его параметров.*

***Ключевые слова:*** *хеш-функция, хаотическое отображение, распараллеливание, безопасность, NIST STS.*

## Введение

Одним из аспектов функциональной безопасности и живучести информационных и информационно-управляющих систем является обеспечение достоверности информации, недопущение как предумышленного, так и непреднамеренного ее разрушения или искажения. Эта задача в современных информационных системах может решаться в том числе и с помощью хеш-функций.

В последнее время в качестве одного из альтернативных подходов к конструированию сжимающих функций алгоритмов хеширования все чаще предлагается использовать достижения теории динамического хаоса. Из исследований в данном направлении следует выделить предложенный в публикации [1] метод построения хеш-функции на основе хаотических отображений с переменными параметрами и параллельной организации вычислений. Большой практический интерес к этому методу обусловлен его способностью в значительной мере компенсировать низкую эффективность расчета траекторий хаотических отображений путем их распараллеливания. Однако детальное исследование, представленное в работах [2] и [3], показало его уязвимость к поиску и обнаружению близких коллизий, а так же коллизий первого и второго рода. В работе [4] был предложен вариант дальнейшего развития и усовершенствования метода, позволяющий устранить выявленные в работах [2, 3] недостатки, а также повысить эффективность вычислений.

Несмотря устранение выявленных недостатков, вопрос соответствия предложенного метода требованиям, выдвигаемым к хеш-функции [5] (в частности устойчивость к возникновению коллизий), остается открытым. Ответ на него могут дать только детальные и всесторонние исследования метода с привлечением широкого круга экспертов. В то же время разработаны и получили широкое применение статистические методы экспресс-оценки «качества» (статистической безопасности) хеш-функций. В частности, наибольшее распространение и признание в мировом научном сообществе получил пакет статистических тестов, разработанный Национальным институтом стандартов и технологий (NIST) США.

Ниже будет приведены результаты тестирования исходной [1] и усовершенствованной [4] версии метода построения хеш-функции с помощью указанного пакета, выработаны рекомендации по его дальнейшему развитию и применению.

## Методика тестирования хеш-функции пакетом статистических тестов NIST Statistical Test Suite (STS)

Пакет тестов NIST STS был разработан отделом безопасности компьютерных технологий NIST. Полное описание пакета, тестов, рекомендации по выбору методики исследований и интерпретации результатов приведены в публикации [6]. Последняя на момент исследований версия 2.2.1 пакета тестирования доступна по ссылке [7]. Применительно к исследуемой хеш-функции пакет позволяет определить насколько она является статистически безопасной и устойчивой к возникновению коллизий.

Всего пакет содержит 15 тестов. Однако часть из них являются сложными и фактически выборки с последовательностями проходят 188 тестов. Общая структура отдельно взятого теста для каждой последовательности имеет следующий вид [6]:

1. Выдвигаются нулевая гипотеза $H_0$ (предположение о том, что данная двоичная последовательность Sq случайна) и альтернативная гипотеза $H_a$ (последовательность неслучайна).

2. По последовательности Sq по соответствующим тесту алгоритмам рассчитывается значение статистики теста $S$.

3. С использованием специальной функции и значения статистики теста рассчитывается значение вероятности $P \in [0,1]$, которая суммирует силу доказательств против нулевой гипотезы.

4. Значение вероятности P сравнивается с уровнем значимости, выбираемым, как правило, в интервале $\alpha \in [0.001, 0.01]$ и который отражает вероятность ошибок первого рода (неправильного отвергания «хорошей» последовательности). Если $P < \alpha$, то гипотеза $H_0$ отвергается (принимается альтернативная гипотеза $H_a$). В противном случае принимается гипотеза $H_0$ (последовательность «случайна»).

В соответствии с рекомендациями NIST тестирование генераторов последовательностей (в рассматриваемом случае – алгоритмов хеширования) с применением пакета тестов в общем случае должно содержать ряд этапов:

1. Выбор генератора последовательностей. Для оценки внесенных в работе [4] усовершенствований и их влияния на повышение безопасности метода построения хеш-функций в рамках тестирования при построении генератора были выбраны его исходная [1] и усовершенствованная [4] версия.

2. Формирование наборов последовательностей для тестирования. Для исходной версии метода были сформированы 6 выборок из 100 последовательностей длиной по 1000064 бит в каждой, т.е. каждая выборка содержала 1000064*100 бит. Выборки формировались путем вычисления хеш значения сообщений длиной по 256 байт. Три выборки формировались для псевдослучайных наборов сообщений (генерируемых линейным конгруэнтным генератором из библиотеки на языке C) с различной точностью промежуточных вычислений в каждой выборке (8, 16 и 24 бита точности). Еще три для «экстремального» варианта исходных сообщений, сформированных из последовательности нулей с двумя последними байтами, работающими в режиме счетчика и различающихся точностью промежуточных вычислений в каждой выборке (8, 16 и 24 бита точности). Усовершенствованная хеш-функция была представлена 36 выборками по 100 последовательностей в каждой. Помимо описанных выше вариантов формирования последовательностей исследовалось также поведение хеш-функции в режиме генератора с различной глубиной распараллеливания (варианты 4, 2 и 1 потока) и различной длиной формируемого хеша (128 и 256 бит).

3. Тестирование выборок пакетом. Для тестирования применялись все 15 (188) тестов пакета с предлагаемыми по умолчанию параметрами. Порядок тестирования последовательностей из выборок был описан выше.

4. Оценка значений вероятности P. Анализ сформированных значений P позволяет вскрыть конкретные дефекты тестируемых последовательностей и хеш-функции в целом.

5. Оценка прохождения теста (ов) последовательностями в выборках путем сравнения значений вероятности P с выбранным уровнем значимости. На этом этапе также вычисляется процент последовательностей в выборке, прошедших тест.

6. Интерпретация полученных результатов. NIST предлагает два основных подхода (которые, однако не являются исчерпывающими) по оценке качества последовательностей и прохождения теста (ов) всей выборкой (выборками). Первый основан на оценке равномерности распределения значений вероятности P для последовательностей из выборки, второй – на оценке процента прошедших тест последовательностей из выборки. При представлении результатов исследований примем второй подход в качестве основного. Для успешного прохождения каждого отдельного теста всей тестируемой выборкой, в соответствии с выбранными по умолчанию параметрами тестирования, необходимо чтобы процент «хороших» (прошедших тест) последовательностей в выборке был не менее 96%.

В целом при оценке последовательностей и интерпретации результатов могут быть сделаны следующие выводы:

1. Тестирование не показало отклонений от случайности.

2. Тестирование четко указывает на отклонение от случайности.

3. Тестирование безрезультатно.

## Результаты тестирования исходной версии хеш-функции пакетом NIST STS

Тестирование проводилось в соответствии с приведенной выше методикой. Было сформировано 6 выборок для различных режимов функционирования хеш-функции по сто последовательностей в каждой. Обобщенные результаты тестирования представлены в табл. 1. В таблице приведено количество успешно пройденных тестов каждой выборкой (как в абсолютном, так и процентном представлении) в соответствии с выбранным правилом. Кроме того, в табл. 1 для более полного отражения зависимости качества последовательностей от параметров их формирования приведен средний процент «хороших» (успешно прошедших тест) последовательностей, полученный путем усреднения процента прошедших тест последовательностей по всем тестам для каждой выборки. Хотя данная характеристика не является формализованной и не описана в публикации [6] она позволяет более точ-

но оценивать влияние параметров хеш-функции (режимов формирования последовательностей) на характеристики статистической безопасности. Так, например, при псевдослучайном исходном тексте и 16-ти битной точности вычислений сформированная выборка успешно прошла все 188 тестов, в то время как при 24-х битной точности организации вычислений – успешно пройдено только 187 тестов. Эти результаты могут быть неправильно интерпретированы, как свидетельствующие о том, что при 16-ти битной точности вычислений могут быть получены более высокие характеристики статистической безопасности. Однако, средний процент «хороших» последовательностей показывает обратное, а один не пройденный тест для 24-х битной точности вычислений может рассматриваться как флуктуация результатов тестирования в рамках статистической погрешности. Такой вывод является обоснованным в рамках общей модели хеш-функции (метода ее построения).

Таблица 1

Результаты тестирования исходного метода построения хеш-функции

| Вариант последовательности | | Результат тестирования | |
|---|---|---|---|
| Исходный текст | Точность вычислений (бит) | Количество и % успешных тестов | Средний % «хороших» последовательностей |
| Псевдослучайный | 8 | 183 (97.3%) | 97.2% |
| | 16 | 188 (100%) | 99% |
| | 24 | 187 (99.5%) | 99.1% |
| Регулярный (счетчик) | 8 | 20 (10.6%) | 13.7% |
| | 16 | 21 (11.2%) | 17.7% |
| | 24 | 105 (55.9%) | 78.2% |

Результаты тестирования, приведенные в табл. 1, четко указывают на явные и значительные отклонения сгенерированных исходной версией хеш-функции последовательностей от случайных. Интерпретируя полученные результаты применительно к хеш-функции можно сделать вывод о ее низкой статистической безопасности и высокой склонности к возникновению коллизий, в частности:

– исключительное влияние на характеристики хеш-функции имеет точность организации промежуточных вычислений. В случае использования псевдослучайного исходного текста при формировании хеша точность вычислений в 8 бит является недостаточной. Для варианта регулярного исходного текста даже 24-х битная точность вычислений не позволяет добиться приемлемых результатов. Можно ожидать, что при дальнейшем росте точности организации вычислений (например, 32 или 48 бит) будут получены относительно высокие характеристики статистической безопасности хеш-функции. Однако такая мера значительно сузит сферу применения функции, усилит требования к аппаратным платформам, снизит характеристики вычислительной эффективности, и в то же время не позволит устранить недостатки, выявленные в работе [2];

– относительно высокие результаты тестирования с псевдослучайным исходным текстом, сформированным с помощью линейного конгруэнтного генератора, обусловлены в первую очередь хорошими статистическими свойствами самого генератора [8]. В то же время, при достаточной точности организации вычислений хеш-функция не искажает и не вносит отклонений от случайности в результаты теста. Это может свидетельствовать о вцелом правильном подходе к конструированию хеш-функции;

– низкие результаты тестирования с регулярным исходным текстом, сформированным с помощью последовательного счетчика, свидетельствуют о значительных изъянах в структуре хеш-функции, недостаточном «лавинном эффекте» и высокой склонности к возникновению близких коллизий.

## Результаты тестирования усовершенствованной версии хеш-функции пакетом NIST STS

Как и для исходной версии хеш-функции, тестирование проводилось в соответствии с приведенной выше методикой. Было сформировано 36 выборок для различных режимов функционирования хеш-функции по сто последовательностей в каждой. Обобщенные результаты тестирования представлены в табл. 2.

Формат таблицы соответствует формату представления результатов тестирования для исходной версии хеш-функции.

Следует отметить, что верхний предел глубины распараллеливания выбирался исходя из максимального количества блоков данных, на которые разбивалось исходное сообщение (длиной 256 байт). Так, для 256-битной версии хеш-функции предел глубины распараллеливания ограничивался значением 3 (три вектора данных, один вектор соответствовал исходному тексту, второй – дополнению, третий – вектору $\vec{c}$ , см. [4]).

Для 128-ми битной версии предел глубины распараллеливания ограничивался значением 4 (че-

тыре вектора данных, два вектора соответствовали исходному тексту, третий – дополнению, четвертый – вектору $\vec{c}$, см. [4]). Серым цветом в таблице выделены выборки с параметрами формирования, соответствующими тестированным выборкам для исходной версии хеш-функции.

Таблица 2

Результаты тестирования усовершенствованного метода построения хеш-функции

| Вариант последовательности | | | | Результат тестирования | |
|---|---|---|---|---|---|
| Исходный текст | Длина хеша (бит) | Точность вычислений (бит) | Глубина распараллеливания | Количество и % успешно пройденных тестов | Средний % «хороших» последовательностей |
| Псевдослучайный | 128 | 8 | 4 | 188 (100%) | 98,7% |
| | | | 2 | 184 (97,9%) | 98,5% |
| | | | 1 | 97 (51,6%) | 79,5% |
| | | 16 | 4 | 188 (100%) | 99,0% |
| | | | 2 | 188 (100%) | 99,1% |
| | | | 1 | 137 (72,9%) | 90,3% |
| | | 24 | 4 | 187 (99,5%) | 99,1% |
| | | | 2 | 187 (99,5%) | 98,9% |
| | | | 1 | 187 (99,5%) | 99,1% |
| | 256 | 8 | 3 | 183 (97,3%) | 98,6% |
| | | | 2 | 188 (100%) | 99,0% |
| | | | 1 | 113 (60,1%) | 82,7% |
| | | 16 | 3 | 187 (99,5%) | 99,0% |
| | | | 2 | 187 (99,5%) | 99,1% |
| | | | 1 | 132 (70,2%) | 84,8% |
| | | 24 | 3 | 188 (100%) | 99,0% |
| | | | 2 | 187 (99,5%) | 99,1% |
| | | | 1 | 186 (98,9%) | 99,0% |
| Регулярный (счетчик) | 128 | 8 | 4 | 28 (14,9%) | 5,6% |
| | | | 2 | 28 (14,9%) | 8,2% |
| | | | 1 | 28 (14,9%) | 9,5% |
| | | 16 | 4 | 48 (25,5%) | 91,7% |
| | | | 2 | 45 (23,9%) | 90,5% |
| | | | 1 | 28 (14,9%) | 50,3% |
| | | 24 | 4 | 38 (20,2%) | 90,2% |
| | | | 2 | 180 (95,7%) | 97,0% |
| | | | 1 | 181 (96,3%) | 97,2% |
| | 256 | 8 | 3 | 12 (6,4%) | 12,7% |
| | | | 2 | 20 (10,6%) | 16,2% |
| | | | 1 | 21 (11,2%) | 15,5% |
| | | 16 | 3 | 95 (50,5%) | 93,3% |
| | | | 2 | 53 (28,2%) | 89,3% |
| | | | 1 | 64 (34,0%) | 64,9% |
| | | 24 | 3 | 117 (62,2%) | 94,0% |
| | | | 2 | 117 (62,2%) | 93,9% |
| | | | 1 | 102 (54,3%) | 93,7% |

Результаты тестирования, приведенные в табл. 2, указывают на некоторые отклонения последовательностей, сгенерированных усовершенствованной версией хеш-функции, от случайных. В публикации [6] приведен перечень причин, по которым выборки могут не пройти тестирование. Их перечень приведен ниже.

1. Некорректная реализация пакета статистических тестов. Хотя пакет тестирования был откомпилирован при проведении тестирования без каких либо правок в исходном коде и проверен на эталонных последовательностях, поставляемых вместе с тестом, это не гарантирует отсутствие ошибок и некорректностей в коде. Так в процессе тестирование наблюдались сообщения об ошибках, связанных с выходом рассчитываемых величин за пределы значимости. Хотя такие ошибки наблюдались только на некоторых выборках (как правило, при низкой точности вычислений), они могли оказать определенное влияние на общие результаты тестирования.

2. Несовершенство статистических тестов. К сожалению, несмотря на универсализм тестов, пакет не может дать однозначный ответ о качестве последовательностей для всего широкого класса задач, где он востребован. Даже для позиционируемых институтом NIST в качестве «хороших» (эталонных) генераторов псевдослучайных чисел результаты тестирования пакетом NIST STS последовательно-

стей, сгенерированных с их помощью, показывают наличие отклонений «случайности» [8]. Поэтому интерпретировать результаты тестирования (как положительные, так и негативные) следует с определенной осторожностью, особенно для задач, в которых оценка качества делается с помощью пакета тестирования косвенно (в частности, при анализе хеш-функций).

В публикации [8] можно найти результаты тестирования наиболее изученных и признанных «хорошими» генераторов (в том числе, встроенных в пакет). Так для генератора SHA-1, квадратичного конгруэнтного, на основе шифра DES и генератора стандарта ANSI X9.17 (на основе шифра Тройной-DES) результаты показывают неполное прохождение всех тестов. А процент прохождения ими тестов сопоставим с процентом прохождения тестирования для усовершенствованной версии рассматриваемой хеш-функции при некоторых параметрах (как для псевдослучайного, так и для регулярного исходного текста).

3. Некорректная имплементация генератора последовательностей. Применительно к рассматриваемой хеш-функции и метода ее построения можно сделать вывод о ряде дефектов при построении конкретных реализаций хеш-функций. Так, ожидалось, что с уменьшением глубины распараллеливания характеристики статистической безопасности будут улучшаться [4]. Однако для ряда выборок наблюдается поведение противоположное ожидаемому. В частности, этот эффект наиболее заметен при формировании выборок с 16-ти битной точностью организации вычислений, а также при использовании версии хеш-функции с 256-ти битным хешем. В то же время для выборок с 8-ми и 24-х битной точностью и 128-ми битной длиной хеша результаты в целом соответствуют ожидаемым.

Следует отметить, что имплементация генератора на основе хеш-фукнции была произведена в строгом соответствии с описанным в публикации [4] методом, а в качестве конкретной реализации хеш-функции применялись предложенные в этой же публикации и публикации [1] параметры метода. В частности в качестве базовых преобразований были выбраны представители класса кусочно-линейных отображений, инициализирующие значения соответствовали предложенным в работе [1] значениям. Коррекция инициализирующих значений для каждой итерации блочной функции осуществлялась в соответствии с предложенным в работе [4] вариантом (комбинацией операцией «исключающее или» промежуточного хеша и подготовленных инициализирующих значений).

Дальнейший анализ показал, что именно выбор «слабых» инициализирующих значений и несовершенство механизма (способа) их коррекции на каждой итерации блочной функции сжатия для выбранных базовых преобразований привели к отклонению результатов тестирования от ожидаемых. Указанные недостатки в имплементации хеш-функции приводили к ухудшению характеристик ее статистической безопасности, причем наиболее заметно это проявлялось именно на малой точности вычислений. При дальнейшем увеличении точности потеря качества компенсировалась улучшением статистических характеристик уже за счет повышения точности организации вычислений.

Однако отметим, что анализу подвергался именно метод построения хеш-функции, а не конкретная его реализация, а выбор базовых преобразований, инициализирующих значений и порядка (способа, механизма) их коррекции на каждой итерации блочной функции сжатия относится к конкретной реализации метода (являются его параметрами). Таким образом, полученные результаты с одной стороны лишь подтверждают корректность метода построения хеш-функций, а с другой – указывают на необходимость более скрупулезного подхода к выбору его параметров при построении конкретных реализаций.

Кроме того, результаты тестирования также указывают на необходимость внедрения некоторых ограничений при построении хеш-функций и выбора сфер ее применения. Так, глубина распараллеливания не должна превышать число блоков (векторов) данных, на которые разбивается исходное сообщение (с учетом дополнения), поскольку в этом случае хеш-функция, построенная на основе усовершенствованного метода, практически сводится к исходной ее версии (хоть и с несколько более высокими характеристиками). Также результаты тестирования указывают на необходимость внесения ограничений на выбор точности организации вычислений.

4. Некорректный программный код реализации генератора последовательностей. Хотя авторами исследований были предприняты максимально полные меры для достижения достоверности и воспроизводимости результатов исследований при написании соответствующего программного кода, не исключается наличий ошибок в коде (например, связанных с некорректным приведением типов данных при организации вычислений), приводящих к искажению и ухудшению результатов тестирования.

5. Недостатки математических функций, используемых для вычисления значений вероятности P. Качественное программно-математическое обеспечение имеет решающее значение для обеспечения достоверной аппроксимации результатов тестирования. NIST в своих рекомендациях указывает, что для ряда тестов возможны ошибки при вычислении P-

значений, связанные с трудностями организации численных приближений. Поэтому NIST рекомендует использовать по возможности в качестве параметров тестов значения по умолчанию.

6. Неправильный выбор параметров тестов. NIST признает, что результаты тестирований рядом тестов крайне чувствительны к входным параметрам и на практике они не будут давать достоверные результаты для всего диапазона допустимых значений. В целом значения параметров по умолчанию оптимизированы для выборок с длинными последовательностями (не менее 1 миллиона бит для большинства тестов). В то же время для ряда генераторов при соблюдении всех рекомендаций использование этих параметров может не приводить к корректным результатам.

В качестве примера можно привести генератор на основе хеш-функции SHA-1, который, не смотря на свои общепризнанные высокие статистические характеристики, не проходит при рекомендуемых параметрах по умолчанию один из тестов непересекающихся шаблонов, а также имеет некоторые предостережения при выборе параметров энтропийного теста.

Можно предположить, что варьируя параметрами тестов, можно достичь более высоких результатов для отдельных тестов, однако эти меры значительно не изменят картину результатов исследований.

В целом, можно констатировать факт, что пакет тестирования NIST STS допускает некоторые флуктуации результатов тестирования, обусловленные описанными выше причинами. Повысить точность аппроксимации результатов тестирования модно путем увеличения числа последовательностей в выборках – однако этот путь требует значительных затрат вычислительных ресурсов при организации тестирования и как правило применяется при оценке генераторов, претендующих на широкое распространение и использование для решения прикладных задач (например, при оценке кандидатов на замещение стандарта хеширования). Для экспресс-оценки и получения общей картины статистической безопасности рассматриваемой хеш-функции приведенные результаты являются достаточными.

В публикации [6] указывается, что количество и процент успешно пройденных тестов должны оцениваться непосредственно авторами исследований исходя из специфики исследуемого генератора. Опираясь на результаты, опубликованные в работе [8], можно предварительно утверждать, что для задачи оценки хеш-функции приемлемыми будут результаты тестирования с числом пройденных тестов от 180 из 188 без значительных отклонений процента «хороших» последовательностей для каждого отдельного теста.

## Выводы

Таким образом, обобщая результаты тестирования и их оценки применительно к хеш-функции можно сделать следующие общие выводы:

– тестирование показало в целом правильность предпринятых мер по усовершенствованию метода построения хеш-функции на основе хаотических отображений с переменными параметрами и параллельной организацией вычислений, предложенных в работе [4]. Однако, для каждой конкретной реализации метода (построенной на его основе хеш-функции) требуются отдельные исследования по выбору оптимальных (рациональных) инициализирующих значений для выбранных базовых преобразований и наиболее эффективного механизма (способа) их коррекции на каждой итерации блочной функции сжатия, которые позволили бы получать уникальные траектории отображений максимальной длины;

– как и для исходной версии хеш-функции, значительное влияние на характеристики ее усовершенствованной версии имеет точность организации промежуточных вычислений. Однако эта зависимость имеет менее требовательный к вычислительным ресурсам характер. Так, уже при 24-х битной точности вычислений для версии 128 битной хеш-функции с базовыми преобразованиями на основе кусочно-линейных отображений могут быть получены приемлемые характеристики статистической безопасности. А при оптимизации механизма формирования (коррекции) инициализирующих значений – ожидаются высокие характеристики и значительное усиление «лавинного эффекта» и при 16-ти битных вычислениях для 128 битной версии и возможно для 256-ти битной. Следует отметить, что увеличение длины хеша приводит к более высоким требованиям к уникальности траекторий отображений, в силу большей рассчитываемой их длины;

– параметры хеш-функции должны подбираться таким образом, чтобы глубина распараллеливания не превышала число блоков (векторов) данных, на которые разбивается исходное сообщение (с учетом дополнения), так как в противном случае хеш-функция практически сводится к исходной ее версии. Это правило несколько ограничивает область применения хеш-функций, построенных на основе усовершенствованного метода (длина обрабатываемого сообщения должна быть не менее $L$ байт, где $L$ – длина хеша в битах). Можно предположить, что дополнение исходного текста (сообщения) до требуемого размера позволит обойти это ограничение, однако эффективность этой меры требует дополнительных исследований;

– с учетом специфики усовершенствованного метода, а также сформулированных ограничений на выбор его параметров, хеш-функции, построенные

на его основе, могут применяться для расчета контрольных сумм (дайджестов) достаточно больших блоков данных (длиной не менее L байт) на мульти процессорных (мультиядерных) платформах. Основной областью применения хеш-функций, построенных на основе метода, является решение задач проверки и подтверждения целостности данных в сфере обеспечения функциональной безопасности и живучести информационных и информационно-управляющих систем, а также (в совокупности с методами криптографической защиты информации) – обеспечение и проверка их подлинности в сфере информационной безопасности. Примерами таких классов задач могут быть задачи контроля целостности файловой структуры хранилищ данных в вычислительно-информационных сетях или задачи формирования электронной цифровой подписи.

## Список литературы

*1. Yantao Li. Parallel Hash function construction based on chaotic maps with changeable parameters [Text] / Yantao Li, Di Xiao, Shaojiang Deng, Qi Han, Gang Zhou // Neural Computing and Applications – 2011. – Vol. 20, №8. – P 1305-1312.*

*2. Антонов А.В. Анализ уязвимостей структуры хеш-функции на основе хаотических отображений с переменными параметрами и параллельной организации вычислений [Текст] / А.В. Антонов // Системи управління, навігації та зв'язку. – К.: ДП «ЦНДІ НІУ», 2012. – Вип. 2(22). – С. 157-162.*

*3. Антонов А.В. Анализ уязвимостей реализации в цифровых вычислительных системах хеш-функции на основе хаотических отображений с переменными параметрами и параллельной организации вычислений [Текст] / А.В. Антонов // Збірник наукових праць Харківського університету Повітряних Сил. – Х.: ХУ ПС, 2012. – Вип. 4(33). – С. 123-129.*

*4. Антонов А.В. Развитие метода построения хеш-функции на основе хаотических отображений с переменными параметрами и параллельной организацией вычислений [Текст] / А.В. Антонов // Системи озброєння та військова техніка. – 2012. – № 2(30). – С. 111-117.*

*5. Al-Kuwari S. Cryptographic Hash Functions: Recent Design Trends and Security Notions [Text] / Saif Al-Kuwari, James Davenport, and Russell Bradford // Proceedings of Inscrypt '10. – Science Press of China, 2010. – P. 133-150 (Режим доступа к полной он-лайн версии доклада: http : www/eprint.iacr.org/2011/565).*

*6. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22, Rev. 1a [Text] / Technology Administration, U.S. Department of Commerce. – Washington: National Institute of Standards and Technology. – 2010. – P. 131.*

*7. NIST Statistical Test Suite, version 2.1.1 [Электронный ресурс] = sts-2.1.1 : [программный код]. – 42 МБ. – National Institute of Standards and Technology, Technology Administration, U.S. Department of Commerce. – 1 электрон. архив (zip) – Систем. требования: Pentium 3; 1 MHz ; 1 Gb RAM ;128 Mb Video ;Windows XP/2000/Vista/7 with Visual Studio 2005 ; Ubuntu Linux with gcc ; Apple MacBook Pro with gcc. – Режим доступа к архиву: http://csrc.nist.gov/groups/ST/toolkit/rng/documents/sts-2.1.1.zip*

*8. Кузнецов А.А. Исследование статистической безопасности генераторов псевдослучайных чисел / А.А. Кузнецов, Р.В. Королев, Ю.Н. Рябуха // Системи обробки інформації. – Х.: ХУ ПС, 2008. – Вип. 3(70). – С. 79-82.*

*Поступила в редколлегию 8.08.2012*

**Рецензент:** д-р техн. наук, проф. П.Ю. Костенко, Харьковский университет Воздушных Сил им. И. Кожедуба, Харьков.

## АНАЛІЗ СТАТИСТИЧНОЇ БЕЗПЕКИ МЕТОДУ ПОБУДОВИ ҐЕШ-ФУНКЦІЇ НА ОСНОВІ ХАОТИЧНИХ ВІДОБРАЖЕНЬ ЗІ ЗМІННИМИ ПАРАМЕТРАМИ Й ПАРАЛЕЛЬНОЇ ОРГАНІЗАЦІЇ ОБЧИСЛЕНЬ (ЗА ДОПОМОГОЮ ПАКЕТУ NIST STS)

А.В.Антонов, В.В. Слободянюк

*Пакетом статистичного тестування NIST STS досліджується базовий і вдосконалений метод побудови ґеш-функцій на основі хаотичних відображень зі змінними параметрами й паралельної організації обчислень. Показано, що запропоновані для вдосконалення методу міри дозволяють досягти високих характеристик його статистичної безпеки. За результатами досліджень сформульовані рекомендації із застосування методу й вибору його параметрів.*

***Ключові слова**: геш-функція, хаотичне відображення, розпаралелювання, безпека, NIST STS.*

## STATISTICAL SAFETY ANALYSIS OF METHOD OF CONSTRUCTING OF HASH FUNCTION BASED ON CHAOTIC MAPS WITH CHANGEABLE PARAMETERS AND PARALLEL CALCULATIONS (USING THE NIST STATISTICAL TEST SUITE)

A.V. Antonov, V.V. Slobodyanyuk

*Basic and improved method of constructing of hash function based on chaotic maps with changeable parameters and parallel calculations were investigated using the NIST statistical test suite. It is shown that the actions taken to improve the method can achieve high performance of its statistical security. According to studies made recommendations on the application method and the choice of its parameters.*

***Keywords**: hash function, the chaotic map, paralleling, security, NIST STS .*