АЛГОРИТМ СТОХАСТИЧЕСКОГО ПРЕОБРАЗОВАНИЯ 3DGOST

Иванов М. А.[1], Комаров Т. И.[2], Кондахчан М. А.[3], Стариковский А. В.[4]

DOI: 10.21681/2311-3456-2024-5-28-33

Аннотация. *Перспективным направлением при решении задач защиты информации является использование стохастических методов, основным результатом применения которых является внесение непредсказуемости в работу компьютерной системы и средств ее защиты.*

Целью данной работы *является обоснование возможности эффективного использования 64-разрядных алгоритмов стохастического преобразования данных, хорошо зарекомендовавших себя в прошлом.*

Метод достижения цели *заключается в использовании архитектуры Куб.*

Полученные результаты: *представлен 3D алгоритм нелинейного преобразования данных, ориентированный на реализацию с использованием гетерогенных суперкомпьютерных технологий. Тестирование алгоритма в режиме генерации псевдослучайных чисел показало его статистическую безопасность.*

Ключевые слова: *генератор псевдослучайных чисел, стохастическое преобразование, непредсказуемость, стохастические методы защиты информации.*

Введение

Важнейшей характеристикой любой компьютерной системы (КС), независимо от ее сложности и назначения, является безопасность обрабатываемой в ней информации. Перспективным направлением при решении задач защиты информации (ЗИ) является использование стохастических методов, основанных на использовании генераторов псевдослучайных чисел (ГПСЧ). Главным результатом применения стохастических методов обработки данных является внесение непредсказуемости в работу КС и средств ее защиты [1, 2].

3D алгоритм стохастического преобразования

Тенденцией последних лет является массовое появление 2D и 3D алгоритмов стохастического преобразования, в частности криптоалгоритмов, ориентированных на реализацию с использованием суперкомпьютерных технологий [2–17]. В [17] предлагается новый 3D алгоритм стохастического преобразования, названный 3DGOST, который может использоваться при построении нелинейной функции ГПСЧ (функции выхода в случае использования режима CTR (Counter Mode) или функции обратной связи в случае использования режима OFB (Output Feedback)).

В данной работе предлагается Light-Weight версия алгоритма 3DGOST, при создании которой главной целью являлось построение нелинейного многораундового преобразования, имеющего повышенное быстродействие за счет упрощения процедуры формирования раундовых ключей и подключей.

В совокупности признаков предлагаемого алгоритма используются следующие термины:

Стохастическое преобразование (Stochastic Transformation) – непредсказуемое преобразование данных; примером стохастического преобразования может являться криптографическое преобразование;

Генератор псевдослучайных чисел (Pseudo-Random Number Generator) – генератор последовательности чисел, статистически не отличимой от последовательности случайных чисел с равномерным законом распределения; наиболее жесткие требования предъявляются к ГПСЧ, ориентированным на решение задач ЗИ;

Ключ (Key) – секретный параметр стохастического преобразования, представляет собой двоичную информацию, известную только законному пользователю;

Подключ (SubKey) – часть ключа;

Раунд (Round) – последовательность шагов, образующих одну итерацию итеративного (многораундового) преобразования;

Раундовый ключ (RoundKey) – ключевая информация, использующаяся при выполнении одного раунда преобразования, существует два способа формирования раундовых ключей: раундовый ключ может являться частью секретного ключа (пример –

1 Иванов Михаил Александрович, доктор технических наук, профессор, главный научный сотрудник ИИКС НИЯУ МИФИ, Москва, Россия. E-mail: maivanov@mephi.ru
2 Комаров Тимофей Ильич, доцент кафедры компьютерных систем и технологий (№12) НИЯУ МИФИ, Москва, Россия. E-mail: tikomarov@mephi.ru
3 Кондахчан Микаэл Арсенович, студент кафедры компьютерных систем и технологий (№12) НИЯУ МИФИ, Москва, Россия. E-mail: mikarkon@gmail.com
4 Стариковский Андрей Викторович, Руководитель проектов Государственного университета управления, Москва, Россия. E-mail: av_starikovskiy@guu.ru

ГОСТ 28147-89), последовательность раундовых ключей может получаться в результате работы процедуры разворачивания исходного ключа (KeyExpansion) (пример – американский стандарт AES);

Раундовый подключ (SubRoundKey) – часть раундового ключа;

Двоичный вектор – некоторая последовательность нулевых и единичных бит, например, (01101010); двоичный вектор разрядности n может быть интерпретирован как элемент конечного поля $GF(2^n)$;

Замена (Substitution) – операция, выполняемая над двоичным вектором $i \in GF(2^n)$, при этом результат операции равен содержимому ячейки с индексом i таблицы замен размерности $n \times 2^n$;

Перемешивание (Mix) – операция, выполняемая над двоичным вектором разрядности m, результат разрядности m которой зависит от всех входных бит и от их взаимного расположения.

Базовое стохастическое преобразование – n раундов произвольного блочного преобразования, работающего с 64-разрядными блоками данных (примеры таких преобразований – ГОСТ 26147-89, Магма (ГОСТ Р 34.12-2015)). Величина n выбирается таким образом, чтобы соответствующее число раундов преобразования обеспечивали полное рассеивание и перемешивание информации (например, для ГОСТ 26147-89 $n \geq 6$).

Суть предлагаемого алгоритма проиллюстрирована рис. 1–4. На рис. 1 показаны блок данных (состояние S или ключ K), принцип разделения блока данных на слои параллельно плоскостям $y0z$, $x0z$, $x0y$, отдельные слои L_{xi}, L_{yi}, L_{zi} блока данных; $i = 0, 1, ..., 7$. На рис. 2 приведена последовательность выполнения преобразования, показаны входной преобразуемый блок данных, выходной преобразованный блок данных; раундовые ключи $KL_{x0}, KL_{x1}, ..., KL_{x7}$ первого раунда; раундовые ключи $KL_{y0}, KL_{y1}, ..., KL_{y7}$ второго раунда; раундовые ключи $KL_{z0}, KL_{z1}, ..., KL_{z7}$ третьего раунда; слои блока данных, которые преобразуются в первом раунде; слои блока данных, которые преобразуются во втором раунде; слои блока данных, которые преобразуются в третьем раунде.

Основные идеи, лежащие в основе предлагаемого алгоритма:

- Представление 512-разрядного состояния S (State) алгоритма, т.е. входных и выходных блоков данных, всех промежуточных результатов преобразований в виде кубического массива бит 8 × 8 × 8 (рис. 1);
- Определение понятия слоя данных (L_{ji}, Layer) – квадратного массива битов 8 × 8, при этом $S = L_{x0} \| L_{x1} \| ... \| L_{x7} = L_{y0} \| L_{y1} \| ... \| L_{y7} = L_{z0} \| L_{z1} \| ... \| L_{z7}$; где $\|$ – операция конкатенации; L_{xi} – слои данных, параллельные плоскости $y0z$; L_{yi} – слои данных, параллельные плоскости $x0z$; L_{zi} – слои данных, параллельные плоскости $x0y$; $i = 0, 1, ..., 7$; $j \in \{x, y, z\}$;
- Представление 512-разрядного ключа K в виде трехмерного массива 8 × 8 × 8 бит (рис. 1);
- Определение понятия слоя ключа (KL_{ji}, KeyLayer), представляемого в виде двухмерного массива 8 × 8 бит, при этом $K = KL_{x0} \| KL_{x1} \| ... \| KL_{x7} = KL_{y0} \| KL_{y1} \| ... \| KL_{y7} = KL_{z0} \| KL_{z1} \| ... \| KL_{z7}$, где $\|$ – операция конкатенации, $j \in \{x, y, z\}$;
- Деление куба данных или ключа на слои параллельно плоскостям $y0z$, $x0z$, $x0y$;
- Двадцатичетырехкратное (по числу слоев) выполнение операции перемешивания слоя Mix с использованием шестираундового базового стохастического преобразования; операция Mix перемешивания слоя данных L_{ji} реализована в виде 6 итераций сети Фейстеля, обеспечивающих полное рассеивание и перемешивание информации. В каждой итерации используются таблицы замен размерностью 4 × 8 × 256 (в случае использования четырех 8-разрядных блоков замены) или 8 × 4 × 16 (в случае использования восьми 4-разрядных блоков замены);
- В качестве двадцати четырех 64-разрядных раундовых ключей преобразования слоев данных используются соответствующие 64-разрядные слои ключа; восемь слоев $KL_{x0}, KL_{x1}, ..., KL_{x7}$ используются в первом раунде преобразования состояния S в качестве раундовых ключей при преобразовании слоев данных соответственно $L_{x0}, L_{x1}, ..., L_{x7}$; восемь слоев $KL_{y0}, KL_{y1}, ..., KL_{y7}$ используются во втором раунде преобразования состояния S в качестве раундовых ключей при преобразовании слоев данных соответственно $L_{y0}, L_{y1}, ..., L_{y7}$; восемь слоев $KL_{z0}, KL_{z1}, ..., KL_{z7}$ используются в третьем раунде преобразования состояния S в качестве раундовых ключей при преобразовании слоев данных соответственно $L_{z0}, L_{z1}, ..., L_{z7}$;
- каждый 64-разрядный слой ключа KL_{ji} суть конкатенация 32-разрядных подключей k_1 и k_2, которые при выполнении шести раундов базового стохастического преобразования используются в следующей последовательности $k_1, k_2, k_2, k_1, k_1, k_2$ (шаг вперед, шаг назад и шаг вперед).

Последовательность преобразования (рис. 2):

1) По входному блоку данных M разрядностью 512 бит формируется блок данных S (состояние алгоритма) той же разрядности в соответствии с выражением $S := M$, после этого выполняются три раунда преобразования состояния S соответственно параллельно плоскостям $y0z$, $x0z$, $x0y$.

2) При выполнении преобразований первого раунда состояние S делится на восемь слоев данных $L_{x0}, L_{x1}, ..., L_{x7}$ параллельно плоскости $y0z$; каждый слой L_{xi} условно представляется в виде квадратного массива битов 8 × 8, после чего подвергается преобразованию Mix, затем преобразованные слои данных объединяются в преобразованный блок данных S.
3) При выполнении преобразований второго раунда состояние S делится на восемь слоев данных $L_{y0}, L_{y1}, ..., L_{y7}$ параллельно плоскости $x0z$; каждый слой L_{yi} условно представляется в виде квадратного массива битов 8 × 8, после чего подвергается преобразованию Mix, затем преобразованные слои данных объединяются в преобразованный блок данных S.
4) При выполнении преобразований третьего раунда состояние S делится на восемь слоев данных $L_{z0}, L_{z1}, ..., L_{z7}$ параллельно плоскости $x0y$, каждый слой L_{zi} условно представляется в виде квадратного массива битов 8 × 8, после чего подвергается преобразованию Mix, затем преобразованные слои данных объединяются в преобразованный блок данных S.

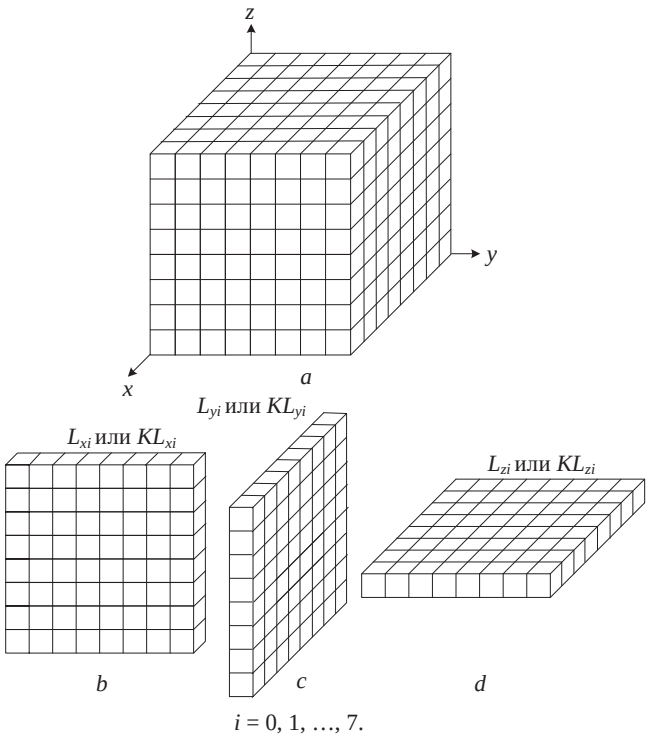

Рис. 1. Стохастическое преобразование 3DGOST: ***a*** *– блок данных (состояние S или ключ K);* ***b*** *– принцип разделения блока данных на слои параллельно плоскости $y0z$, слой L_{xi};* ***c*** *– принцип разделения блока данных на слои параллельно плоскости $x0z$, слой L_{yi};* ***d*** *– принцип разделения блока данных на слои параллельно плоскости $x0y$, слой L_{zi} блока данных; $i = 0, 1, ..., 7; j \in \{x, y, z\}$.*

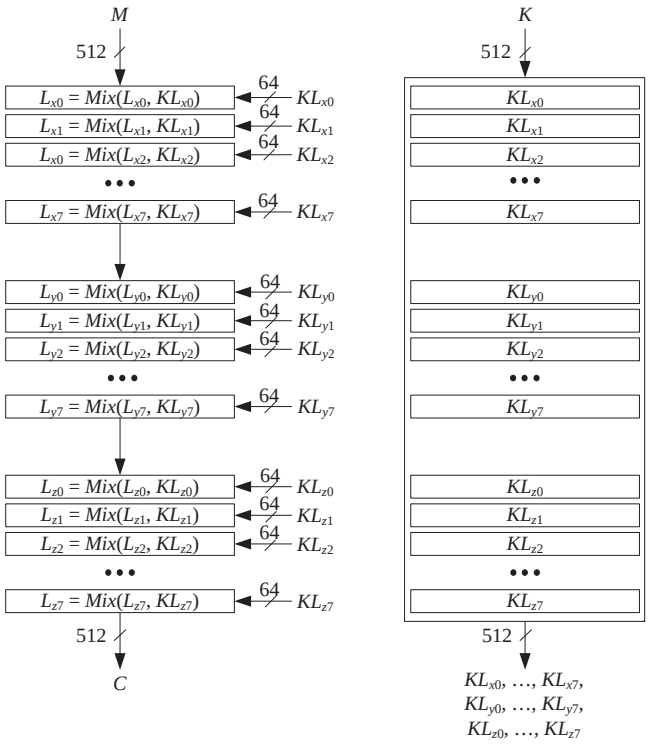

Рис. 2. Последовательность преобразования 3DGOST

Рис. 2 демонстрирует также принцип деления ключа на раундовые ключи. Показан ключ K; слои KL_{xi} ключа, которые являются раундовыми ключами первого раунда преобразования; слои KL_{yi} ключа, которые являются раундовыми ключами второго раунда преобразования; слои KL_{zi} ключа, которые являются раундовыми ключами третьего раунда преобразования.

На рис. 3 показан пример реализации базового стохастического преобразования (БСП) на основе шестираундовой сети Фейстеля.

Каждая из 6 итераций БСП (преобразования Mix) (рис. 3, а) может являться, например, раундом ГОСТ 28147-89, предполагающим деление входного 64-разрядного блока данных на левую L (Left) и правую R (Right) половины, последовательное выполнение операций $T = (R + SK) \bmod 2^{32}$, $T = S(T)$, $T = \mathrm{Rol}^{11}(T)$, $T = T$ XOR L, $L = R$, $R = T$, объединение новых значений L и R в преобразованный 64-разрядный блок данных, где T (Temporary) – временная переменная, SK (SubKey) – 32-разрядный подключ, $S()$ – 32-разрядная операция замены (Substitution), Rol^{11} (RotateLeft) – операция циклического сдвига на 11 разрядов влево 32-разрядного входного слова, XOR – 32-разрядная операция поразрядного сложения по модулю два (рис. 3, б).

На рис. 3 показаны RL_{ji} и LL_{ji} – соответственно младшая (Right) и старшая (Left) половины входного слоя данных L_{ji}; $RL_{ji}{}^*$ и $LL_{ji}{}^*$ – соответственно младшая и старшая половины преобразованного

слоя данных L_{ji}^*; F – раундовая функция, DI (DataIn) – входные 32-разрядные данные, k – 32-разрядный раундовый подключ (k_1 или k_2), DO (DataOut) – выходные 32-разрядные данные, Sub (Substitution) – 32-разрядный блок замены (S-блок).

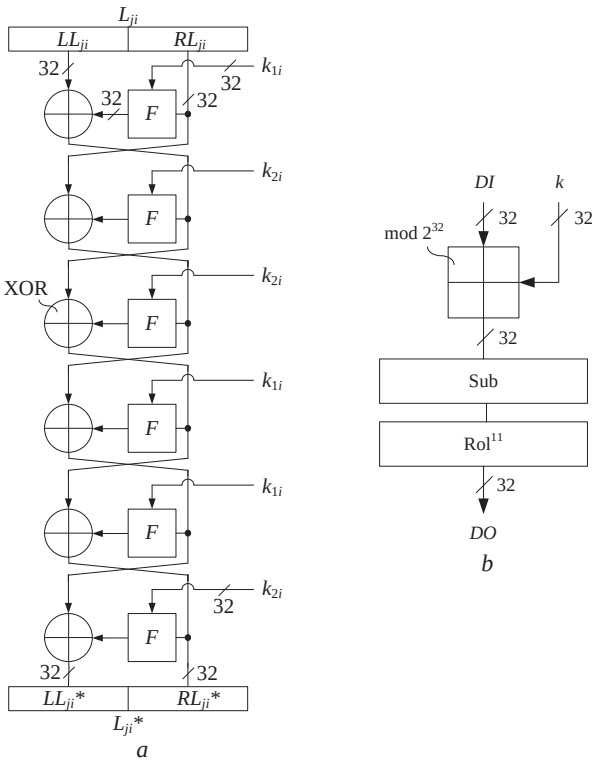

Рис. 3. Вариант реализации преобразования Mix слоев Lij:
***a** – схема базового стохастического преобразования на основе шестираундовой сети Фейстеля; **b** – вид функции F, которая была специфицирована в ГОСТ 28147-89.*

На рис. 4 показан принцип деления раундовых ключей KL_{xi}, KL_{yi} и KL_{zi} на подключи k_1 и k_2. RKL_{ji} и LKL_{ji} – соответственно младшая (Right) и старшая (Left) половины входного слоя ключа KL_{ji}.

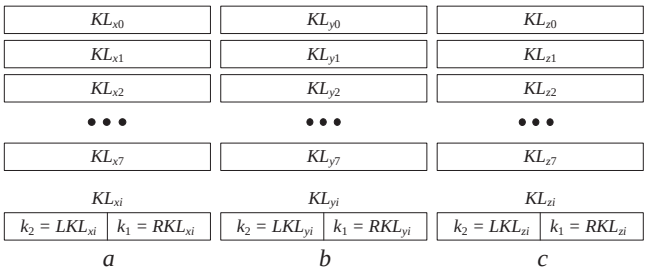

Рис. 4. Принцип деления раундовых ключей KL_{xi} (a), KL_{yi} (b) и KL_{zi} (c) на подключи k_1 и k_2.

Результаты статистического тестирования

Результаты статистического тестирования по методике НИСТ [18, 19] генератора псевдослучайных чисел, построенного по двухступенчатой схеме Counter Mode, где вторая ступень – это преобразование 3DGOST, показаны на рис. 5.

Файлов в запросе:	1000
Пропущено из-за размера:	0
Пропущено из-за периода:	0
Тестировалось:	1000
Из них прошло тесты:	
Проверка 0 и 1:	992
Проверка 0 и 1 в подпосл:	989
Проверка несц. серий:	976
Проверка сцеп. серий:	972
Проверка дырок:	990
Проверка дырок в подпосл.:	988
Проверка непер. шаблонов:	1000
Проверка пер. шаблонов:	986
Проверка частот:	987
Проверка интервалов:	960
Проверка перестановок:	912
Проверка на монотонность:	910
Универсальный тест Маурера:	985
Проверка рангов:	986
Проверка кум. сумм:	985
Проверка случ. отклон.:	927

a

Файлов в запросе:	1000
Пропущено из-за размера:	0
Пропущено из-за периода:	0
Тестировалось:	1000
Из них прошло тесты:	
Проверка 0 и 1:	990
Проверка 0 и 1 в подпосл:	992
Проверка несц. серий:	979
Проверка сцеп. серий:	977
Проверка дырок:	995
Проверка дырок в подпосл.:	988
Проверка непер. шаблонов:	1000
Проверка пер. шаблонов:	978
Проверка частот:	998
Проверка интервалов:	964
Проверка перестановок:	903
Проверка на монотонность:	907
Универсальный тест Маурера:	988
Проверка рангов:	993
Проверка кум. сумм:	985
Проверка случ. отклон.:	914

b

Рис. 5. Результаты тестирования ГПСЧ с функцией выхода на основе преобразования 3DGOST:
***a** – разрядность выходной последовательности 512 бит;*
***b** – разрядность выходной последовательности 8 бит.*

Заключение

Предложенное 3D стохастическое преобразование ориентировано на реализацию с использованием гетерогенных суперкомпьютерных технологий. Очевидно, что в пределах каждого раунда преобразования все восемь слоев состояния могут быть обработаны параллельно, поэтому применение, например, технологии CUDA [20, 21] позволит существенно упростить процесс разработки ПО. Предлагаемое решение позволит продлить жизнь многим качественным 64-разрядным криптоалгоритмам, не «дотягивающим» до требуемого сейчас 256-битного уровня безопасности для блочных шифров и 512-битного уровня безопасности для криптографических хеш-функций.

Нелинейное трехмерное многораундовое преобразование данных имеет повышенное быстродействие за счет максимального упрощения процедуры формирования раундовых ключей и подключей. Строго говоря, никакого формирования вообще нет. Раундовые ключи – это 24 слоя исходного ключа, имеющего кубическую структуру 8 × 8 × 8, показанную на рис. 1. Раундовые подключи каждого из 24-х шестираундовых преобразований Mix – это младшая и старшая половины раундовых ключей, которые используются по принципу «шаг вперед, шаг назад и шаг вперед». В результате нелинейное трехмерное многораундовое преобразование данных может использоваться в условиях ограниченных ресурсов, так как является Light-Weight алгоритмом, т.е. для решения задач защиты информации в RFID-системах и Интернете вещей.

Тестирование преобразование показало статистическую безопасность алгоритма.

Литература

1. Иванов М. А. Стохастические методы защиты информации. – Всероссийская научно-техническая конференция «Кибернетика и информационная безопасность» (КИБ 2023). Сборник научных трудов, Москва, 2023, с. 42-43.
2. Иванов М. А., Скитев А. А., Стариковский А. В. Классификация генераторов псевдослучайных чисел, ориентированных на использование в задачах защиты информации. (2016). [Электронный ресурс]. https://www.aha.ru/~msa/papers11.pdf (Дата обращения: 10.06.2024).
3. Joan Daemen, Lars Knudsen, Vincent Rijmen. The Block Cipher Square. (1998). [Электронный ресурс]. https://www.ime.usp.br/~rt/cranalysis/square.pdf (Дата обращения: 07.06.2016).
4. Joan Daemen, Vincent Rijmen. The Design of Rijndael. AES — The Advanced Encryption Standard. Springer-Verlag, Berlin, Heidelberg, NewYork, London, Paris, Tokyo, Hong Kong, Barcelona, Budapest, 2001, 253 p.
5. Jorge Nakahara Jr. 3D: A Three-Dimensional Block Cipher. In: Franklin M.K., Hui L.C.K., Wong D.S. (eds.) CANS 2008. LNCS, vol. 5339, pp. 252–267. Springer, Heidelberg, 2008.
6. P. Barreto, V. Rijmen. The WHIRLPOOL Hashing Function. (2003). [Электронный ресурс]. https://cryptospecs.googlecode.com/svn/trunk/hash/specs/whirlpool.pdf (дата обращения: 10.06.2024).
7. Keccak sponge function family. Main document. Guido Bertoni, Joan Daemen, Michael Peeters, Gilles Van Assche. (2008) [Электронный ресурс]. https://keccak.noekeon.org/Keccak-main-2.1.pdf (дата обращения: 10.06.2024).
8. G. Bertoni, J. Daemen, M. Peeters, G. Van Assche. Keccak specifications. (2009). [Электронный ресурс]. https://keccak.noekeon.org/Keccak-specifications-2.pdf (дата обращения: 10.06.2024).
9. R. Benadjila, O. Billet, H. Gilbert, G. Macario-Rat, T. Peyrin, M. Robshaw, Y. Seurin. SHA-3 proposal: ECHO. (2009). [Электронный ресурс]. https://crypto.rd.francetelecom.com/echo/doc/echo_description_1-5.pdf (Дата обращения: 10.06.2024).
10. P. Gauravaram, L. R. Knudsen, K. Matusiewicz, F. Mendel, C. Rechberger, M. Schlaffer and S. S. Thomsen. Grøstl – a SHA-3 candidate. (2011). [Электронный ресурс]. https://perso.uclouvain.be/fstandae/ source_codes/hash_atmel/specs/groestl.pdf (Дата обращения: 10.06.2024).
11. Eli Biham and Orr Dunkelman. The SHAvite-3 Hash Function. (2009). [Электронный ресурс]. https://ehash.iaik.tugraz.at/uploads/f/f5/Shavite.pdf (Дата обращения: 10.06.2024).
12. Информационная технология. Криптографическая защита информации. Функция хеширования. ГОСТ Р 34.11-2012. – Москва, Стандартинформ, 2012.
13. GOST 34.12-2018. Information Technology. Cryptographic Information Defense. Block Ciphers, 2018. Moscow: Standartinform.
14. Ivanov M. A., Vasilyev N. P., Chugunkov I. V. Three-dimensional data stochastic transformation algorithms for hybrid supercomputer implementation. (2012). [Электронный ресурс]. https://2012.nscf.ru/Tesis/Ivanov.pdf (Дата обращения: 10.06.2024).
15. Using Sequential and Parallel Composition for Stochastic Data Processing/ Ivanov M. A., Kozyrsky B. L., Komarov T. I., et.al. – Proceedings of The Radio-Electronic Devices and Systems for the Infocommunication Technologies (REDS-2013), Moscow, Russia, May 22-23, 2013, pp.144–148.
16. Three New Methods of Stochastic Data Transformaion/M. A. Ivanov, I. V. Matveychikov, A. A. Skitev, et. al. – Proceedings of The Radio-Electronic Devices and Systems for the Infocommunication Technologies (REDS-2016), Moscow, Russia, May 25-26, 2016, pp.351–355.
17. Иванов М. А., Стариковский А. В., Щустова Л. И. Новая жизнь старого ГОСТа: переход от одномерной версии к 3D. – REDS: Телекоммуникационные устройства и системы, 2017, Т. 7, № 4, с. 488–491.
18. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. A. Rukhin, J. Soto, J. Nechvatal, et.al. NIST Special Publication 800-22, Revision 1a. 2010.
19. Чугунков И. В. Методы и средства оценки качества генераторов псевдослучайных последовательностей, ориентированных на решение задач защиты информации. Учебное пособие. – М.: НИЯУ МИФИ, 2012.
20. Боресков А. В., Харламов А. А. Основы работы с технологией CUDA. М.: ДМК Пресс, 2011.
21. CUDA C++ Programming Guide. Release 12.5. NVIDIA, 2024.

3DGOST STOCHASTIC TRANSFORMATION ALGORITHM

Ivanov M. A.[5], Komarov T. I.[6], Kondakhchan M. A.[7], Starikovsky A. V.[8]

Abstract. *A promising direction in solving information security problems is the use of stochastic methods, the main result of which is the introduction of unpredictability into the operation of a computer system and network security tools.*

The purpose of this work *is to substantiate the possibility of effective use of 64-bit stochastic data transformation algorithms, which have proven themselves well in the past.*

The method to achieve *the goal is to use the Cube architecture.*

Results obtained: *a 3D algorithm for nonlinear data transformation is presented, oriented towards implementation using heterogeneous supercomputer technologies. Testing the algorithm in pseudorandom number generation mode showed its statistical safety.*

Keywords: *pseudorandom number generator, stochastic transformation, unpredictability, stochastic methods of information security.*

5 Mikhail A. Ivanov, Dr. Sc. (Eng), Professor, Chief Researcher, Institute of Computer Systems, National Research Nuclear University MEPhI, Moscow, Russia. E-mail: maivanov@mephi.ru

6 Timofey I. Komarov, Associate Professor, Department of Computer Systems and Technologies (No12), National Research Nuclear University MEPhI, Moscow, Russia. E-mail: tikomarov@mephi.ru

7 Mikael A. Kondakhchan, Student, Department of Computer Systems and Technologies (No12), National Research Nuclear University MEPhI, Moscow, Russia. E-mail: mikarkon@gmail.com

8 Andrey V. Starikovsky, Project Manager, State University of Management, Moscow, Russia. E-mail: av_starikovskiy@guu.ru

References

1. *Ivanov M. A. Stohasticheskie medody zashchity infomacii. – Vserossijskaj nauchno-technicheskaya Conferentsya «Kibernetica i informatcionnaya bezopasnost» (KIB 2023). Sbornik nauchnih trudov, Moskva, 2023, c. 42-43. (in Russian).*
2. *Ivanov M. A., Skitev A. A., Starikovskij A. V. Klassifikatciya generatorov psevdosluchainyh chisel orientirovannyh na ispolsovanie v zadachah zachity informatsii. (2016). [Electronic resource]. https://www.aha.ru/~msa/papers11.pdf (Date Views: 10.06.2024). (in Russian).*
3. *Joan Daemen, Daemen, Joan, Lars Knudsen, Vincent Rijmen. The Block Cipher Square. (1998). [Electronic resource]. https://www.ime.usp.br/~rt/cranalysis/square.pdf (Date Views: 07.06.2016).*
4. *Joan Daemen, Vincent Rijmen. The Design of Rijndael. AES – The Advanced Encryption Standard. Springer-Verlag, Berlin, Heidelberg, NewYork, London, Paris, Tokyo, Hong Kong, Barcelona, Budapest, 2001, 253 p.*
5. *Jorge Nakahara Jr. 3D: A Three-Dimensional Block Cipher. In: Franklin M. K., Hui L. C. K., Wong D. S. (eds.) CANS 2008. LNCS, vol. 5339, pp. 252–267. Springer, Heidelberg, 2008.*
6. *P. Barreto, V. Rijmen. The WHIRLPOOL Hashing Function. (2003). [Electronic resource]. https://cryptospecs.googlecode.com/svn/trunk/hash/specs/whirlpool.pdf (Date Views: 10.06.2024).*
7. *G. Bertoni, J. Daemen, M. Peeters, G. Van Assche. Keccak specifications. (2009). [Electronic resource]. https://keccak.noekeon.org/Keccak-specifications-2.pdf (Date Views: 10.06.2024).*
8. *Keccak sponge function family. Main document. Guido Bertoni, Joan Daemen, Michael Peeters, Gilles Van Assche. [Electronic resource]. http://keccak.noekeon.org/Keccak-main-2.1.pdf (Date Views 07.06.2016)*
9. *R. Benadjila, O. Billet, H. Gilbert, G. Macario-Rat, T. Peyrin, M. Robshaw, Y. Seurin. SHA-3 proposal: ECHO. (2009). [Electronic resource]. http://crypto.rd.francetelecom.com/echo/doc/echo_description_1-5.pdf (Date Views: 10.06.2024).*
10. *P. Gauravaram, L. R. Knudsen, K. Matusiewicz, F. Mendel, C. Rechberger, M. Schlaffer and S. S. Thomsen. Grøstl – a SHA-3 candidate. (2011). [Electronic resource]. https://perso.uclouvain.be/fstandae/ source_codes/hash_atmel/specs/groestl.pdf (Date Views: 10.06.2024).*
11. *Eli Biham and Orr Dunkelman. The SHAvite-3 Hash Function. (2009). [Electronic resource]. https://ehash.iaik.tugraz.at/uploads/f/f5/Shavite.pdf (Date Views: 10.06.2024).*
12. *GOST R 34.11-2012. Information Technology. Cryptographic Information Defense. Hash Finction.– Moscow, Standartinform, 2012. (in Russian).*
13. *GOST 34.12-2018. Information Technology. Cryptographic Information Defense. Block Ciphers. –Moscow, Standartinform, 2018. (in Russian).*
14. *Ivanov M. A., Vasilyev N. P., Chugunkov I. V. Three-dimensional data stochastic transformation algorithms for hybrid supercomputer implementation. (2012). [Electronic resource]. https://2012.nscf.ru/Tesis/Ivanov.pdf (Date Views: 10.06.2024).*
15. *Using Sequential and Parallel Composition for Stochastic Data Processing/ Ivanov M. A., Kozyrsky B. L., Komarov T. I., et.al. – Proceedings of The Radio-Electronic Devices and Systems for the Infocommunication Technologies (REDS-2013), Moscow, Russia, May 22-23, 2013, pp.144–148.*
16. *Three New Methods of Stochastic Data Transformaion/M. A. Ivanov, I. V. Matveychikov, A. A. Skitev, et. al. – Proceedings of The Radio-Electronic Devices and Systems for the Infocommunication Technologies (REDS-2016), Moscow, Russia, May 25-26, 2016, pp.351–355.*
17. *Ivanov M. A., Starikovskiy A. V., Shchustova L. I. Novaya zhizn' starogo GOSTa: perekhod ot odnomernoy versii k 3D. – REDS: Telekommunikatsionnyye ustroystva i sistemy, 2017, T. 7, № 4, s. 488–491. (in Russian).*
18. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. A. Rukhin, J. Soto, J. Nechvatal, et.al. NIST Special Publication 800-22, Revision 1a. 2010.*
19. *Chugunkov I. V. Metody i sredstva otsenki kachestva generatorov psevdosluchaynyh posledovatel'nostey. Uchebnoe posobie. – M.: NRNU MEPhI, 2012. (in Russian).*
20. *Boreskov A. V., Harlamov A. A. Osnovy raboty s tehnologiey CUDA. M.: DMK Press, 2011. (in Russian).*
21. *CUDA C++ Programming Guide. Release 12.5. NVIDIA, 2024.*