УДК 004.056.55

Л.К. Бабенко, Д.В. Голотин

ОБ ОСНОВНЫХ ОСОБЕННОСТЯХ ФУНКЦИОНИРОВАНИЯ И РЕАЛИЗАЦИИ ПОТОЧНОГО ШИФРА TRIVIUM*

Тенденция массового перехода от интернета персональных компьютеров к интернету вещей (Internet of Things, IoT) ставит новые задачи по обеспечению надежности и безопасности работы сетей. Беспроводная среда передачи, отсутствие инфраструктуры, большой поток данных – эти факторы позволяют злоумышленнику достаточно легко провести анализ сети на уязвимости и реализовать атаку. Одним из основных средств защиты информации для IoT является использование криптографических алгоритмов. В связи с тем, что многие приспособления в сети мобильны и зачастую имеют небольшие размеры, существуют общие ограничения на ресурсы времени и памяти. Эти ограничения распространяются и на криптографические схемы, открывая новое направление разработки и исследования алгоритмов малоресурсной криптографии. Построение эффективных аппаратных реализаций специально разработанных алгоритмов шифрования является актуальным и определено задачами малоресурсной криптографии. Цель данной статьи – создать программную модель поточного шифра малоресурсной криптографии Trivium, провести ее исследование и ознакомить специалистов с особенностями построения и функционирования этого алгоритма, с полученными характеристиками. Программная модель должна быть наглядной и иметь возможности для применения ее в образовательном процессе. Статья посвящена рассмотрению структуры, описанию работы алгоритма Trivium, оценки возможностей программно-аппаратной реализации. Структура Trivium представляет собой три сдвиговых регистра с обратной связью, которые совместно формируют псевдослучайную последовательность (ПСП). Сложение по модулю два одноименных битов открытого текста и ПСП определяет биты криптограммы. Процедура инициализации производится на основе заданного секретного ключа и вектора инициализации. Для изучения и анализа возможностей алгоритма Trivium разработана программная модель, позволяющая наглядно увидеть процессы, возникающие в ходе работы данного шифра. Созданная программная модель может быть использована для обучения студентов основам поточного шифрования на примере алгоритма Trivium. При использовании шифра для реализации задач малоресурсной криптографии необходимо оценить его возможности по аппаратной реализации. С этой целью в статье рассматривается реализация некоторых блоков шифра с использованием программируемых логических схем (ПЛИС) Altera, приводятся фрагменты описания их на языке VHDL, характеристики программной и аппаратной реализации.

Поточное шифрование; алгоритм Trivium; малоресурсная криптография; программная модель; аппаратная реализация; проектирование с использованием ПЛИС.

L.K. Babenko, D.V. Golotin

THE MAIN FEATURES FUNCTIONING AND IMPLEMENTATION STREAM CIPHER TRIVIUM

The trend of mass transfer from the Internet PC to Internet of things (Internet of Things, IoT) poses new challenges to ensure the reliability and safety of the networks. Wireless transmission medium, dynamic changing topology, lack of infrastructure, large data stream, - these factors allow an attacker to easily analyze network vulnerabilities and implement an attack. One of the main means of information protection is the use of cryptographic algorithms. Due to the fact that many mobile devices, often have a small size, and there are restrictions on the time and memory resources. These restrictions apply to the cryptographic schemes, opening a new direction «Lightweight cryptography algorithms». Building effective hardware implementations of specially

* Работа выполнена при поддержке грантом РФФИ № 15-07-00595 – а.

developed encryption algorithms is relevant and defined objectives Lightweight cryptography. The purpose of this article - to create a program model stream cipher Lightweight cryptography Trivium, to conduct its investigation and to familiarize professionals with the features of construction and operation of this algorithm, with the characteristics. The programming model is to be visible and be able to apply it in the educational process. The article considers the structure, the description of the algorithm Trivium, assess the feasibility of software and hardware implementation. Trivium structure consists of three shift registers with feedback, which together form a pseudo-random sequence (PRS). Modulo two like bits of plaintext bits and cap defines cryptogram. The initialization procedure is performed based on a predetermined secret key and initialization vector. To study and analyze opportunities Trivium algorithm software model was developed that allows to visualize the processes that occur in the course of this cipher. Software model can be used to teach students the basics of stream encryption algorithm on the example of Trivium. When using encryption to achieve the objectives few resource cryptography is necessary to assess its ability to hardware implementation. To this end, the article discusses the implementation of some of the blocks are encrypted using programmable logic (FPGAs) Altera, are fragments of the description of the language VHDL, the characteristics of software and hardware implementation.

Stream encryption; algorithm Trivium; few resource cryptography; programming model; hardware implementation; design using FPGAs.

Введение. Все прочнее в нашу жизнь входит бесчисленное число всевозможных электронных приспособлений, направленных на улучшение жизни и так или иначе взаимодействующих с интернетом, на смену интернета персональных компьютеров приходит интернету вещей (Internet of Things, IoT), широкое распространение получают беспроводные самоконфигурирующиеся сенсорные сети. Одной из основных проблем эффективного применения таких сетей является использование специальных аппаратно реализованных алгоритмов шифрования – алгоритмов малоресурсной криптографии, обладающих заданными характеристиками по размеру микросхемы, потребляемой энергии, объему оперативной памяти, времени исполнения программы [1–6]. Построение эффективных аппаратных реализаций специально разработанных алгоритмов шифрования – актуально и определено задачами малоресурсной криптографии [7–12].

Ранее нами была рассмотрена аппаратная реализация двух симметричных блочных алгоритмов Present и Clefia [13, 14]. Продолжение развития темы определило цель данной статьи – создать программную модель поточного шифра малоресурсной криптографии Trivium, провести ее исследование и ознакомить специалистов с особенностями построения и функционирования этого алгоритма, с полученными характеристиками программной реализации и основных блоков аппаратной реализации, с перспективами эффективной аппаратной реализации полного алгоритма. Программная модель должна быть наглядной и иметь возможности для применения ее в образовательном процессе.

Краткое описание алгоритма. Структура Trivium представляет собой три сдвиговых регистра общей длинной 288 бит с нелинейной комбинацией прямой и обратной связи, которые совместно формируют псевдослучайную последовательность (бегущий ключ) длиной вплоть до $N \le 2^{64}$ бит. Первый регистр алгоритма Trivium имеет длину 93 бита, второй – 84 бита, третий – 111 бит.

При каждом такте выполнения алгоритма происходит сдвиг вправо на один бит в каждом из битовых регистров, а на выходе генерируется один бит бегущего ключа. Шифрование сообщения производится путем выполнения операции XOR битов сообщения с соответствующими битами бегущего ключа. Расшифрование – путем выполнения операции XOR битов шифра сообщения с соответствующими битами бегущего ключа [15, 16].

На рис. 1 представлен общий вид алгоритма Trivium. Числами представлена позиция бита в регистрах, операция $\otimes$ означает умножение, а операция $\oplus$ сложение по модулю два.

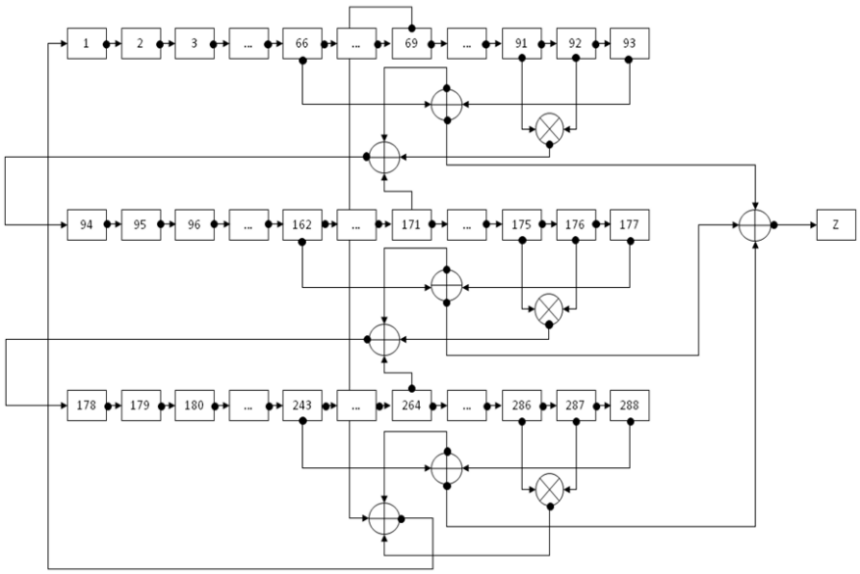

Рис. 1. Общий вид алгоритма Trivium

Для начального заполнения регистров Trivium используются два источника:

1) секретный ключ K;
2) инициализирующий вектор IV.

Размеры ключа и инициализирующего вектора составляют 80 бит.

Работа алгоритма Trivium состоит в следующем. Перед генерацией псевдослучайной последовательности Z, необходимо произвести процедуру инициализации, заключающуюся в том, что в начале регистры Trivium заполняются битами из секретного ключа K и инициализирующего вектора IV. После этой процедуры происходит выполнение цикла длиной в 288*4 раз алгоритмаTrivium, без генерации Z. Последняя процедура необходима для того, чтобы каждый бит начального состояния регистров зависел от каждого бита IV и K. На самом деле это достигается при выполнении двух полных циклов (288*2 раз), остальные два цикла необходимы для усложнения взаимосвязей битов регистра.

На рис. 2 наглядно изображена блок-схема процедуры инициализации начального состояния регистров алгоритма Trivium. В роли W[I] выступают биты регистров.

Как видно из схемы, изменение состояний регистров происходит путём сдвига информации на один бит вправо в каждом из регистров. Последние биты в регистрах используются в сумме по модулю два с некоторыми другими битами для генерации битов, которые перемещаются в начало регистров. Этими битами на блок-схеме являются биты T1, T2 и T3.

После проведения операции инициализации схема Trivium готова приступить к своей главной задаче – генерации последовательности Z.

На рис. 3 изображена блок-схема, показывающая генерацию последовательности.

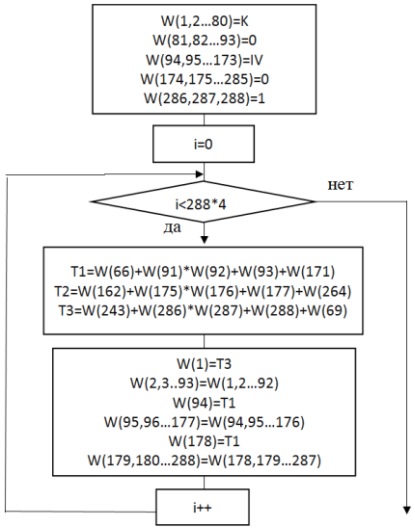

Рис. 2. Блок-схема процедуры инициализации начального состояния регистров

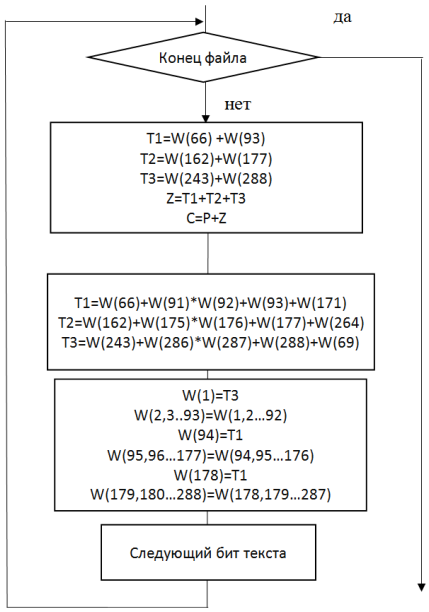

Рис. 3. Блок–схема генерации последовательности Z

В целом ход работы Trivium не слишком изменяется по отношению к процессу инициализации. Единственным отличием является то, что теперь на каждом такте выполнения алгоритма генерируется один бит последовательности Z, который представляет собой сумму по модулю два шести битов регистров (по два с каждого регистра). Номера битов участвующих в генерации Z (как видно из блок-схемы на рис. 3), – 66, 93, 162, 177, 243, 288. На блок-схеме биты открытого текста обозначаются P, а зашифрованного – C.

Полученные результаты. Создана программная модель шифра Trivium, работающая в консольном режиме.

После инициализации программы она взаимодействует с тремя файлами, содержащими закрытый ключ, инициализирующий вектор и документ, который необходимо зашифровать.

По окончании считывания исходных данных программа выполняет непосредственно шифрование путём сложения по модулю два одного байта исходного документа и одного байта псевдослучайной последовательности Z, полученной при помощи алгоритма Trivium, сгенерированной побитно.

Программная модель рассматривает шифруемый документ как двоичный файл, вследствие чего тип файла не является критичным, что позволяет шифровать документы таких форматов, как pdf.

Для оценки качества генерируемой ПСП в состав программной модели встроена возможность использования таких тестов, как [17}:

- тест частоты знаков;
- покер–тест;
- тест проверки серий;
- тест проверки последовательной корреляции;
- тест проверки на монотонность.

Разработанная программа предназначена не только для создания зашифрованных документов открытых текстов, но и позволяет проследить битовые взаимодействия нескольких тактов работы программы.

Шифрование выполняется с достаточно неплохой скоростью, так на шифрование 13,5 Мб, выполненное на компьютере, тактовая частота процессора которого 2,1 ГГц, ушло 168,7 с. Если привести к единице измерения кбит/с, то получим 640 кбит/с. Для сравнения, реализация DES на языке perl (http://www.opennet.ru/base/dev/perl_crypt.txt.html) на процессоре с частотой 1,7 ГГц показывает результат в 300 кбит/с [18].

Переход к аппаратной реализации. Первым шагом для перехода к аппаратной реализации является структурная схема, изображённая на рис. 4.

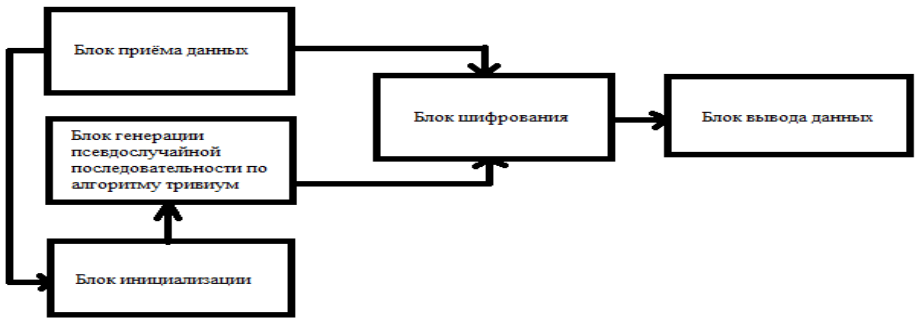

Рис. 4. Структура аппаратной реализации шифрования

Следующий подготовительный шаг к аппаратной реализации – выбор инструментария, при помощи которого будет реализован шифр Trivium.

Для аппаратной реализации будут рассматриваться программируемые логические интегральные схемы (ПЛИС), так как:

- во-первых, при создании логических устройств на ПЛИС разработчик не ограничен в элементной базе потому, что современные ПЛИС содержат библиотеки, в которых есть и простые логические элементы и более сложные, такие как микропроцессоры;
- во-вторых, ПЛИС позволяет многократно самому разработчику корректировать схему, не внося изменений в печатный монтаж;

- в-третьих, использование ПЛИС зачастую способствует уменьшению габаритов аппаратуры по сравнению с их аналогами, реализованными на традиционных БИС.

Язык программирования на ПЛИС выбран VHDL, как наиболее апробированный и зарекомендовавший себя с лучшей стороны [19–21].

Пример описания блока инициализации на языке VHDL:

```
if flag='0' then  // переменная flag нужна для того, чтобы обозначить очерёдность выполнения кода
i:=0;
while i<288 loop  // в этом цикле инициализируется начальное состояние регистров.
if i<80 then Sud(i)<=Bud1(i);
elsif i<93 then Sud(i)<='0';
elsif i<173 then Sud(i)<=Bud2(i-93);
elsif i<285 then Sud(i)<='0';
else Sud(i)<='1';
end if;
i:=i+1;
if i>287 then flag:='1'; // переход к выполнению следющей части алгоритма
end if;
end loop;
if flag=1 then // проверяем что выполняется данная часть кода
j:=0;
t1:=Sud(65)xor(Sud(90)and Sud(91))xor sud(92)Xor Sud(170);  // высчитываем сдвиговые биты t1,t2 ,t3
t2:=Sud(161)xor(Sud(174)and Sud(175))xor sud(176)xor sud(263);
t3:=Sud(242)xor (sud(285)and sud(286))xor Sud(287) xor Sud(68);
Sud<= Sud(286 downto 177)& t2& Sud(175 downto 93)& t1 & Sud(91 downto 0) & t3; // делаем сдвиги в регистрах,
j:=j+1;
if j>1151 then flag:=2; // проверяем было ли совершено 4*288 раз выполнения сдвигов, в целях инициализации регистра.
end if;
end if;
```

Разработаны основные блоки аппаратной реализации алгоритма Trivium в среде проектирования Qurtus II. Рассчитана производительность аппаратной реализации, составляющая 20М бит/с.

Заключение. Итогом данной работы является программная модель шифра Trivium, которая может быть использована, прежде всего, для обучения студентов при изучении современных поточных шифров малоресурсной криптографии. Разработанная программа предназначена не только для создания зашифрованных документов открытых текстов, но и позволяет проследить битовые взаимодействия нескольких тактов работы программы.

Программа также может выводить данные о времени, потраченном на шифрование заданного файла, что позволяет судить об эффективности шифра. Исходя из того, что программа на процессоре частотой 2,1 ГГц имеет скорость 640 кбит/с, что более, чем в 2 раза превосходит скорость шифрования по алгоритму DES на процессоре 1,7 ГГц, можем сделать вывод о том, что разработанная программная реализации шифра Trivium имеет хорошую скорость, а алгоритм является эффективным.

При выполнении работы рассмотрена возможность и разработаны основные блоки аппаратной реализации алгоритма Trivium в среде проектирования Qurtus II. На основе действующей модели рассчитана производительность аппаратной реализации, составляющая 20 Мбит/с.

На следующем этапе работы предполагается доведение разработки до действующего аппаратно реализованного образца, на котором будет возможно производить экспериментальные исследования по оптимизации аппаратных решений шифра малоресурсной криптографии Trivium.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Жуков А.Е.* Легковесная криптография // Вопросы кибербезопасности. – 2015. – № 1 (9). – С. 26-43.
2. *Axel York Poschmann*. Lightweight cryptography Cryptographic Engineering for a PervasiveWorld. Dissertation for the degree Doktor-Ingenieur Faculty of Electrical Engineering and Information Technology Ruhr-University Bochum, Germany, 2013.
3. *Кяжин С.Н., Моисеев А.В.* Криптография в облачных вычислениях: современное состояние и актуальные задачи. Режим доступа: www.pvti.ru/data/file/bit/2013/2013_3/part_15.pdf.
4. *Лоуренс Круз.* Интернет вещей и информационная безопасность. http://www.cisco.com/web/RU/news/releases/txt/2013/03/032813c.html (дата обращения 20.03.2014).
5. *Bogdanov A., Knudsen L.R., Leander G., Paar C., Poschmann A., Matthew Robshaw, Seurin Y., and Vikkelsoe C.* PRESENT: An ultra-lightweight block cipher. In CHS 2007, volume 7427 of Lecture Notes in Computer Science, Springer Verlag, 2007. – P. 450-466.
6. *Panasenko S., Smagin S*. Lightweight Cryptography: Underlying Principles and Approaches. International // Journal of Computer Theory and Engineering. – August 2011. – Vol. 3, No. 4. – P. 516-520.
7. *Canni`ere C.D., Dunkelman O., and M. Knezevic.* KATAN and KTANTAN a Family of Small and Efficient Hardware-Oriented Block Ciphers, CHES 2009, LNCS 5747. – Springer-Verlag, 2009. – P. 272-278.
8. *Julio Cesar Hernandez-Castro1, Pedro Peris-Lopez2, Jean-Philippe Aumasson.* On the Key Schedule Strength of PRESENT. School of Computing, Portsmouth University, UK, Information Security & Privacy Lab, TU-Delft, The NetherlandsNagravisionSA, Cheseaux, Switzerland.
9. The 128-bit Blockcipher CLEFIA. Security and Performance Evaluations. – Revision 1.0, June 1, Sony Corporation. – 2007.
10. *Toru Akishita and Harunaga Hiwatari*. Very Compact Hardware Implementations of the Blockcipher CLEFIA. – Sony Corporation: {Toru.Akishita,Harunaga.Hiwatari}@jp.sony.com.
11. *Shirai T., Shibutani K., Akishita T., Moriai S., and Iwata T.* Hardware Implementations of the 128-bit Blockcipher CLEFIA // Technical Report of IEICE. – 2007. – Vol. 107, No. 141, ISEC2007–49. – P. 29-36, (in Japanese).
12. *Moradi A. Poschmann, S. Ling C. Paar, and H. Wang*. Pushing the Limits:A Very Compact and a Threshold Implementation of AES", EUROCRYPT 2011, LNCS 6632. – Springer-Verlag, 2011. – P. 69-88.
13. *Бабенко Л.К., Беспалов Д.А., Макаревич О.Б., Чесноков Р.Д., Трубников Я.А*. Разработка и исследование программно-аппаратного комплекса шифрования по алгоритму PRESENT для решения задач малоресурсной криптографии // Известия ЮФУ. Технические науки. – 2014. – № 2 (151). – С. 174-180.
14. *Бабенко Л.К., Беспалов Д.А., Макаревич О.Б., Чесноков Р.Д.* Разработка и исследование средств малоресурсной криптографии на примере алгоритмов PRESENT И CLEFIA // Материалы конференции "Информационные технологии в управлении" (ИТУ-2014). – СПб.: ОАО "Концерн "ЦНИИ "Электроприбор", 2014.
15. *C. De Canni`ere & B. Preneel*. TRIVIUM Specifications. eSTREAM, ECRYPT Stream Cipher Project [электронный ресурс]. URL: http://www.ec rypt.eu.org/stream/p3ciphers/trivium/trivium_p3.pdf (дата обращения: 14.02.2014).

16. *Yun Tian, Gongliang Chen, Jianhua Li.* Quavium - A New Stream Cipher Inspired by Trivium // JCP journal of computers. – 2012. – Vol. 7, No. 5. – P. 1278-1283.
17. *Варфоломеев А.А., Жуков А.Е., Пудовкина М.А.* Поточные криптосистемы. Основные свойства и методы анализа стойкости: Учебное пособие. – М.: ПАИМС, 2000. – 272 с.
18. Шифрование данных в Perl программах (crypt perl howto). URL: http://www.opennet.ru/base/dev/perl_crypt.txt.html.
19. *Бабило П.Н.* Основы языка VHDL. – 2-е изд. – М.: Солон–Р, 2002. – 218 с.
20. *Сергиенко А.М.* VHDL для проектирования вычислительных устройств. – М.: Изд-во: ТИД ДС, 2003. – 208 с.
21. VHDL: A logic synthesis approach Hardcover: D. Naylor, S. Jones. Cambridge University Press – July 31, 199. – 324 p.

REFERENCES

1. *Zhukov A.E.* Legkovesnaya kriptografiya [Lightweight cryptography], *Voprosy kiberbezopasnosti* [Cybersecurity Issues], 2015, No. 1 (9), pp. 26-43.
2. *Axel York Poschmann*. Lightweight cryptography Cryptographic Engineering for a PervasiveWorld. Dissertation for the degree Doktor-Ingenieur Faculty of Electrical Engineering and Information Technology Ruhr-University Bochum, Germany, 2013.
3. *Kyazhin S.N., Moiseev A.V.* Kriptografiya v oblachnykh vychisleniyakh: sovremennoe sostoyanie i aktual'nye zadachi [Cryptography in cloud computing: current state and challenges]. Available at: http://www.pvti.ru/data/file/bit/2013/2013_3/part_15.pdf.
4. *Lourens Kruz.* Internet veshchey i informatsionnaya bezopasnost' [The Internet of things and information security]. Available at: http://www.cisco.com/web/RU/news/releases/txt/2013/03/032813c.html (Accessed 20 March 2014).
5. *Bogdanov A., Knudsen L.R., Leander G., Paar C., Poschmann A. Matthew Robshaw, Y. Seurin, and C. Vikkelsoe.* PRESENT: An ultra-lightweight block cipher. In CHS 2007, volume 7427 of Lecture Notes in Computer Science, Springer Verlag, 2007, pp. 450-466.
6. *Panasenko S., Smagin S*. Lightweight Cryptography: Underlying Principles and Approaches. International, *Journal of Computer Theory and Engineering,* August 2011, Vol. 3, No. 4, pp. 516-520.
7. *Canni`ere C.D., Dunkelman O., and M. Knezevic.* KATAN and KTANTAN a Family of Small and Efficient Hardware-Oriented Block Ciphers, CHES 2009, LNCS 5747. Springer-Verlag, 2009, pp. 272-278.
8. *Julio Cesar Hernandez-Castro1, Pedro Peris-Lopez2, Jean-Philippe Aumasson.* On the Key Schedule Strength of PRESENT. School of Computing, Portsmouth University, UK, Information Security & Privacy Lab, TU-Delft, The NetherlandsNagravisionSA, Cheseaux, Switzerland.
9. The 128-bit Blockcipher CLEFIA. Security and Performance Evaluations. Revision 1.0, June 1, Sony Corporation, 2007.
10. *Toru Akishita and Harunaga Hiwatari*. Very Compact Hardware Implementations of the Blockcipher CLEFIA. Sony Corporation: {Toru.Akishita,Harunaga.Hiwatari}@jp.sony.com.
11. *Shirai T., Shibutani K., Akishita T., Moriai S., and Iwata T.* Hardware Implementations of the 128-bit Blockcipher CLEFIA, *Technical Report of IEICE,* 2007, Vol. 107, No. 141, ISEC2007–49, pp. 29-36, (in Japanese).
12. *Moradi A. Poschmann, S. Ling C. Paar, and H. Wang*. Pushing the Limits:A Very Compact and a Threshold Implementation of AES", EUROCRYPT 2011, LNCS 6632. Springer-Verlag, 2011, pp. 69-88.
13. *Babenko L.K., Bespalov D.A., Makarevich O.B., Chesnokov R.D., Trubnikov Ya.A.* Razrabotka i issledovanie programmno-apparatnogo kompleksa shifrovaniya po algoritmu PRESENT dlya resheniya zadach maloresursnoy kriptografii [Software and hardware development and research of encryption algorithm PRESENT for solving problems of the lightweight cryptography], *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences], 2014, No. 2 (151), pp. 174-180.
14. *Babenko L.K., Bespalov D.A., Makarevich O.B., Chesnokov R.D.* Razrabotka i issledovanie sredst maloresursnoy kriptografii na primere algoritmov PRESENT I CLEFIA [Research and development funds metaresources cryptography example algorithms PRESENT AND CLEFIA], *Materialy konferentsii "Informatsionnye tekhnologii v upravlenii" (ITU-2014)* [The proceedings of the conference "Information technologies in management" (IUT-2014)]. St. Petersburg: OAO "Kontsern "TsNII "Elektropribor", 2014.

15. *C. De Canni`ere & B. Preneel*. TRIVIUM Specifications. eSTREAM, ECRYPT Stream Cipher Project [электронный ресурс]. Available at: http://www.ec rypt.eu.org/stream/p3ciphers/ trivium/trivium_p3.pdf (Accessed: 14 February 2014).
16. *Yun Tian, Gongliang Chen, Jianhua Li.* Quavium - A New Stream Cipher Inspired by Trivium, *JCP journal of computers*, 2012, Vol. 7, No. 5, pp. 1278-1283.
17. *Varfolomeev A.A., Zhukov A.E., Pudovkina M.A.* Potochnye kriptosistemy. Osnovnye svoystva i metody analiza stoykosti. Uchebnoe posobie [Stream cryptosystem. Basic properties and methods of strength analysis. Tutorial]. Moscow: PAIMS, 2000, 272 p.
18. Shifrovanie dannykh v Perl programmakh (crypt perl howto). Available at: http://www.opennet.ru/base/dev/perl_crypt.txt.html.
19. *Babilo P.N.* Osnovy yazyka VHDL [The basics of the language VHDL]. 2 nd ed. Moscow: Solon–R, 2002, 218 p.
20. *Sergienko A.M.* VHDL dlya proektirovaniya vychislitel'nykh ustroystv [VHDL for the design of computing devices]. Izd-vo: TID DS, 2003, 208 p.
21. VHDL: A logic synthesis approach Hardcover: D. Naylor, S. Jones. Cambridge University Press, July 31, 199, 324 p.

Статью рекомендовал к опубликованию д.т.н., профессор Н.И. Витиска.

Бабенко Людмила Климентьевна – Южный федеральный университет; e-mail: blk@tsure.ru; 347928, г. Таганрог, ул. Чехова, 2, корпус "И"; тел.: 88634312018; кафедра безопасности информационных технологий; профессор.

Голотин Денис Владимирович – e-mail: ww.golotin@mail.ru; тел.: +79514944226; кафедра безопасности информационных технологий; студент.

Babenko Lyudmila Klimentevna – Southern Federal University; e-mail: blk@tsure.ru; Block “I”, 2, Chehov street, Taganrog, 347928, Russia; phone: +78634312018; the department of information technologies security; professor.

Golotin Denis Vladimirovich – e-mail: ww.golotin@mail.ru; phone: +79514944226; the department of information technologies security; student.

УДК 004.853

С.В. Поликарпов, К.Е. Румянцев, А.А. Кожевников

ИССЛЕДОВАНИЕ ЛИНЕЙНЫХ ХАРАКТЕРИСТИК ПСЕВДОДИНАМИЧЕСКИХ ПОДСТАНОВОК*

Одним из эффективных путей нейтрализации целого пласта методов криптоанализа блочных криптоалгоритмов является применение динамически изменяющихся подстановок. Подход на основе псевдодинамических подстановок PD-sbox *позволяет совместить сильные стороны фиксированных подстановок (высокая скорость работы и эффективность использования вычислительных ресурсов) и динамических подстановок (нейтрализация статистических методов криптоанализа). Для оптимального применения псевдодинамических подстановок требуется детальное исследование их криптографических свойств. Цель исследования – разработать методику определения линейных характеристик псевдодинамических подстановок для дальнейшего подтверждения их целевого применения в блочных криптоалгоритмах. В работе получены выражения для определения линейных свойств псевдодинамических подстановок PD-sbox для двух случаев: 1) когда значения состояния S фиксированы и задаются криптографическим ключом; 2) когда значения состояния S динамически изменяются под воздействием энтропии входной информации и результатов предшествующих преобразований. Первичный анализ выражения позволил сделать вывод, что сама структура псевдодинамической подстановки PD-sbox*

* Работа выполнена на основе гос. задания Минобрнауки РФ № 213.01-11/2014-9.