ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ
СПбГУТ))) Декабрь 2020 Том 8 № 4 ISSN 2307-1303 http://ijitt.ru/
TELECOM IT
December 2020 Vol. 8 Iss. 4 ISSN 2307-1303 http://ijitt.ru/

АППАРАТНО-ПРОГРАММНЫЙ КОМПЛЕКС ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ МИКРОСХЕМ ФЛЭШ-ПАМЯТИ ДЛЯ ПОДГОТОВКИ СИСТЕМЫ ИДЕНТИФИКАЦИИ

С. С. Владимиров*, А. К. Янковский

Санкт-Петербургский государственный университет телекоммуникаций
им. проф. М. А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация
*Адрес для переписки: vladimirov.opds@gmail.com

Аннотация—**Предмет исследования**. Статья представляет варианты реализации аппаратно-программного комплекса параллельной обработки микросхем флэш-памяти для подготовки системы идентификации. **Метод**. Проведен анализ скорости деградирования модулей памяти и определены средние значения времени обработки и количества циклов обработки для одной микросхемы памяти при подготовке идентификатора. **Основные результаты**. Предложены параллельная и последовательно-параллельная схемы обработки микросхем флеш-памяти. **Практическая значимость**. Предлагается применение разработанного комплекса для формирования идентифицирующих микросхем флэш-памяти, предназначенных для проведения статистических исследований методов и протоколов идентификации.

Ключевые слова—идентификация, флэш-память, деградированная память, параллельная обработка.

Информация о статье
УДК 004.7
Язык статьи – русский.
Поступила в редакцию 22.11.2020, принята к печати 23.12.2020.

Ссылка для цитирования: Владимиров С. С., Янковский А. К. Аппаратно-программный комплекс параллельной обработки микросхем флэш-памяти для подготовки системы идентификации // Информационные технологии и телекоммуникации. 2020. Том 8. № 4. С. 60–68. DOI 10.31854/2307-1303-2020-8-4-60-68.

HARDWARE SYSTEM FOR PARALLEL PROCESSING OF FLASH MEMORY CHIPS FOR PREPARING AN IDENTIFICATION SYSTEM

S. Vladimirov*, A. Yankovskiy

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications,
St. Petersburg, 193232, Russian Federation

*Corresponding author: vladimirov.opds@gmail.com

Abstract—**Research subject**. The paper presents options for the implementation of a hardware-software system for parallel processing of flash memory chips for preparing an identification system. **Method**. The analysis of the degradation rate of memory modules is carried out and the average values of the processing time and the number of processing cycles for one memory chip during the preparation of the identifier are determined. **Core results**. Parallel and serial-parallel schemes for flash memory chips processing are proposed. **Practical relevance**. It is proposed to use the developed complex for the creation of identifying flash memory microcircuits intended for statistical research of identification methods and protocols.

Keywords—identification, flash memory, degraded memory, parallel processing.

Article info
Article in Russian.
Received 22.11.2020, accepted 23.12.2020.

For citation: Vladimirov S., Yankovskiy A.: Hardware System for Parallel Processing of Flash Memory Chips for Preparing an Identification System // Telecom IT. 2020. Vol. 8. Iss. 4. pp. 60–68 (in Russian). DOI 10.31854/2307-1303-2020-8-4-60-68.

Введение

Важным элементом современных систем передачи данных является подсистема идентификации сетевых устройств. Важность надежной идентификации оборудования возрастает с увеличением количества используемых в сети устройств и повышением требований к безопасности [1, 2]. В настоящее время самым распространенным способом идентификации является использование уникального идентификатора, заранее записанного в память устройства. Таким идентификатором, например, является MAC-адрес сетевого устройства и другие варианты расширенного уникального идентификатора EUI, прописываемые в память устройств производителем[1] [3]. Однако, такие идентификаторы могут быть легко заменены на программном уровне, а некоторые недобросовестные производители могут использовать один MAC-адрес для целой партии устройств [4, 5, 6]. Другим примером является международный идентификатор мобильного оборудования IMEI, присваиваемый сотовым телефонам [7, 8]. Также, как и MAC-адрес, идентификатор IMEI может быть подделан, а в случае создания поддельного телефона отличить его от оригинала может быть затруднительно даже при прямом контакте, не говоря уже об удаленной идентификации.

[1] Guidelines for Use of Extended Unique Identifier (EUI), Organizationally Unique Identifier (OUI), and Company ID (CID). IEEE, 2017. 19 p.

Перспективным вариантом решения этой проблемы является использование идентификаторов, уникальность которых обеспечивается физическими законами или особенностями аппаратуры, возникающими при ее производстве. К таким способам, в частности, относят идентификацию по особенностям радиосигнала в случае радиопередающего оборудования [9] и идентификацию по особенностям встроенных в устройства микросхем памяти [10, 11, 12]. На кафедре Сетей связи и передачи данных ведется разработка системы однозначной идентификации на основе уникальных свойств микросхем NOR флеш-памяти с интерфейсом SPI, встраиваемых в современные сетевые устройства [13, 14, 15, 16, 17]. Принцип основан на двух утверждениях: во-первых, каждый чип флеш-памяти уникален из-за особенностей его производства [14]; во-вторых, при многократной перезаписи чипа он начинает деградировать и некоторые битовые ячейки памяти, из которых состоит чип, принимают одно неизменное состояние (в случае NOR флеш-памяти это ноль), т. е. становятся так называемыми «бэд-блоками» [12, 13]. В совокупности получается, что каждый чип памяти при деградации будет иметь уникальное расположение («рисунок») бэд-блоков, что позволит однозначно идентифицировать этот чип, а, следовательно, и устройство, в котором этот чип находится. При этом сам рисунок бэд-блоков является уникальным идентификатором устройства [13, 14, 15].

Для статистических исследований предложенного метода требуется большое количество деградировавших чипов памяти, что позволит подтверждать работоспособность разрабатываемых для него протоколов и систем на больших выборках.

Методика принудительной деградации чипа памяти

Принудительная деградация микросхемы флэш-памяти для получения идентификаторов заключается в принудительном заполнении используемого в качестве идентификатора участка памяти значениями 0×00 (все нули) и последующей очистке данной области, которая переводит ячейки памяти в состояние 0×FF (все единицы). Алгоритм этой процедуры показан на рис. 1. Эффект деградации проявляется в застывании нулевых значений бит в ячейке после стирания. Например, деградировавшая байтовая ячейка после стирания может остаться равной 0×F7 вместо 0×FF, т. е. 5 бит ячейки перестает возвращаться в исходное состояние и застывает на нулевом значении.

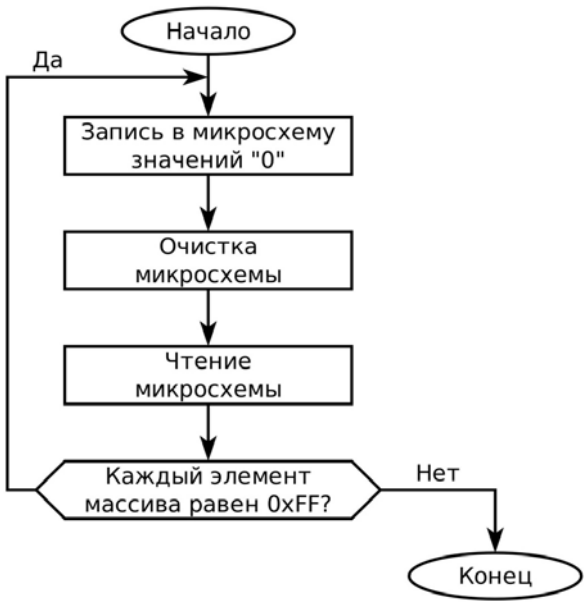

Рис. 1. Алгоритм получения микросхем памяти с деградированными ячейками

Существующие микросхемы имеют определенное гарантируемое производителем число циклов перезаписи, в течении которых появление деградировавших ячеек памяти произойти не может. В среднем, число циклов для памяти NOR-

flash составляет от 100 до 150 тысяч в зависимости от производителя. Например, эксперименты авторов показали, что для микросхем Winbond W25X16 первые деградировавшие битовые ячейки появляются в среднем после 350 тысяч циклов перезаписи и не менее чем после 150 тысяч циклов.

Для оценки скорости деградирования модулей памяти, рассчитаем время, затрачиваемое на один цикл перезаписи одного блока памяти, как сумму времени записи блока и времени его стирания:

$$t_C = t_{WR} + t_{ER}.$$

Время стирания t_{ER} состоит из времени, затрачиваемого на отправку команды и времени ожидания очистки блока памяти t_{BLKE}, определяемого техническими спецификациями микросхемы:

$$t_{ER} = \frac{N_{ER}}{f_{SPI}} + t_{BLKE},$$

где N_{ER} – количество бит в команде стирания блока памяти, а f_{SPI} – тактовая частота интерфейса SPI.

Запись блока памяти происходит постранично, т. е. за один раз записывается участок памяти определенного объёма – страница памяти. Количество страниц в деградируемом (перезаписываемом) блоке памяти назовем n. Размер каждой страницы в байтах обозначим M_P. В выбранных для экспериментов чипах размер страницы равен 256 байт. Каждая запись страницы требует отдельного времени ожидания t_{PP}, определяемого техническими спецификациями микросхемы. Общая формула для вычисления времени записи блока памяти имеет следующий вид:

$$t_{WR} = n \cdot \left(\frac{N_{PP} + M_P}{f_{SPI}} + t_{PP}\right),$$

где N_{PP} – количество бит в команде записи страницы памяти.

Для примера, произведем расчет времени цикла перезаписи t_C для микросхемы AtmelAT26DF161 и перезаписываемого участка памяти, равного одному сектору размером 4096 байт, т. е. состоящего из $n = 16$ страниц памяти по $M_P = 256$ байт каждая. Размер команд для стирания сектора памяти и записи одной страницы одинаков и равен $N_{ER} = N_{PP} = 32$ бита. В формуле (1) приведен расчет времени цикла в секундах для тактовой частоты интерфейса $f_{SPI} = 1$ Мбит/с, а в формуле (2) — для частоты $f_{SPI} = 100$ Мбит/с. Значения времени стирания одного сектора памяти $t_{BLKE} = 0{,}05$ с и время записи одной страницы памяти $t_{PP} = 0{,}0015$ с взяты из официального технического руководства микросхемы и являются средними для данной микросхемы. Дополнительные задержки, вносимые обработкой отдельных команд в микропроцессоре и контроллере SPI, считаем пренебрежимо малыми и не учитываем.

$$t_{C[1\text{Мбит/с}]} = \left(\frac{32}{10^6} + 0{,}05\right) + 16 \cdot \left(\frac{2080}{10^6} + 0{,}0015\right) = 0{,}1073. \quad (1)$$

$$t_{C[100\text{Мбит/с}]} = \left(\frac{32}{10^8} + 0{,}05\right) + 16 \cdot \left(\frac{2080}{10^8} + 0{,}0015\right) = 0{,}0743. \quad (2)$$

Из результатов расчета видно, что в обоих случаях время одного цикла перезаписи является величиной одного порядка и соответствует приблизительно

0,1 мс. Таким образом, время на принудительное деградирование одного сектора флэш-памяти (в среднем 350 тысяч циклов) будет приблизительно составлять от 26000 до 37500 секунд, что примерно соответствует интервалу от 7 до 11 часов. В зависимости от модели микросхемы флэш-памяти это время может быть большим или меньшим, как следует из таблицы. Выбранная для расчета микросхема AtmelAT26DF161 обладает средними характеристиками для микросхем данного типа.

Таблица.

Сравнение скоростных характеристик распространенных моделей NOR флэш-памяти

№	Наименование модуля	t_{BLKE} для блока 4 Кбайта, с	t_{BLKE} для блока 32 Кбайта, с	t_{PP}, с
1	AT26DF161	0,05	0,35	0,0015
2	W25P16	–	0,6	0,0035
3	S25FL032P	0,02	0,5	0,0015
4	W25X10BL	0,03	0,12	0,0007
5	EN25F16	0,15	0,8	0,0015

Разработанные аппаратно-программные комплексы

Изначально, для первичной проверки работоспособности метода был использован аппаратно-программный комплекс, состоящий из микрокомпьютера Raspberry Pi, имеющего встроенный интерфейс SPI, и библиотеки функций pi-spiflash[2] на основе модуля py-spidev[3] языка Python [13]. Этот комплекс удобен для работы с одиночными микросхемами, однако не подходит для получения больших количеств деградированных микросхем из-за высокой стоимости масштабирования комплекса.

Для решения данной проблемы принято решение распараллелелить обработку микросхем. При этом предлагается два подхода: параллельный и последовательно-параллельный.

Параллельный подход основан на использовании отдельного интерфейса SPI на каждую обрабатываемую микросхему памяти. В разработанном программно-аппаратном комплексе в качестве управляющего устройства используется компьютер (ПК) под управлением ОС GNU/Linux. К нему подключаются конвертеры/программаторы USB-SPI на основе микроконтроллера CH341a, которые на сегодня являются самым бюджетным решением такого плана на рынке. Масштабируемость системы достигается установкой множества программаторов на один ПК, при этом современные процессоры ПК обеспечивают параллельную отправку команд за счет работы каждой управляющей программы в отдельном потоке, а наличие на каждой микросхеме отдельного управляющего SPI обеспечивает параллельность обработки этих команд. Для управления конвертерами было написано программное обеспечение ch341prog-extended[4], обеспечивающее возможность параллельной работы с большим числом программаторов CH341a на одном компьютере [16]. Схема данного комплекса приведена на рис. 2.

[2] pi-spiflash. URL: https://github.com/vlad-ss/pi-spiflash
[3] py-spidev. URL: https://github.com/doceme/py-spidev
[4] ch341prog-extended. URL: https://github.com/YAost/ch341prog-extended

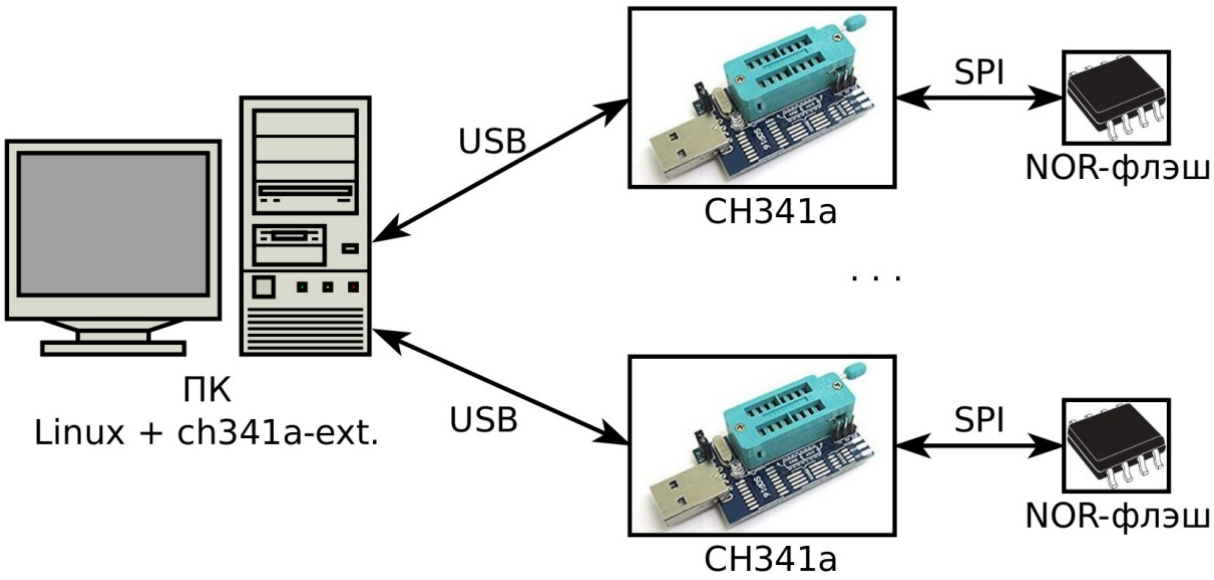

Рис. 2. Схема установки, основанной на параллельной отправке команд,
с использованием программаторов CH341a

Разработанное решение на базе программатора CH341a имеет аппаратное ограничение частоты работы SPI в 1 МГц, вызванное особенностями работы конвертера. Соответственно, быстродействие данного аппаратно-программного комплекса можно оценить в соответствии с формулой (1).

Последовательно-параллельный подход основан на использовании процессора с числом управляющих шин SPI меньшим, чем количество обрабатываемых микросхем. Этот подход основан на том, что время отправки команды на микросхему, значительно меньше времени выполнения команды, и том, что в интерфейсе SPI несколько микросхем могут быть подключены к одним и те же шинам данных и тактирования. Следовательно, за время выполнения команды на одной микросхеме можно отправить команды на другие микросхемы. В параллельной схеме управляющий модуль SPI при этом «простаивает», занимаясь постоянным опросом и ожиданием завершения длительной операции.

Общая схема такого решения представлена на рис. 3. Микросхемы памяти подключаются на общие шины MOSI (master out slave in), MISO (*master in slave out*), SCLK (*serial clock*). Линия CS (*chipselect*) каждой микросхемы подключается к отдельному управляющему GPIO контакту процессора, позволяя отправлять команды микросхемам по отдельности.

В предлагаемой схеме используется следующий алгоритм управления. Управляющий процессор последовательно в цикле отправляет команду записи на каждую из подключенных к интерфейсу SPI микросхем. Далее процессор в цикле опрашивает каждую микросхему, контролируя окончание выполнения команды. После того как любая из микросхем отвечает готовностью к выполнению следующей команды, процессор отправляет в нее команду стирания и продолжает опрос. Таким образом, все подключенные к процессору микросхемы работают параллельно. Если использованный для построения системы деградирования управляющий модуль имеет несколько интерфейсов SPI, то к каждому из них могут быть подключены несколько микросхем, а подбором их количества можно обеспечить максимальную производительность.

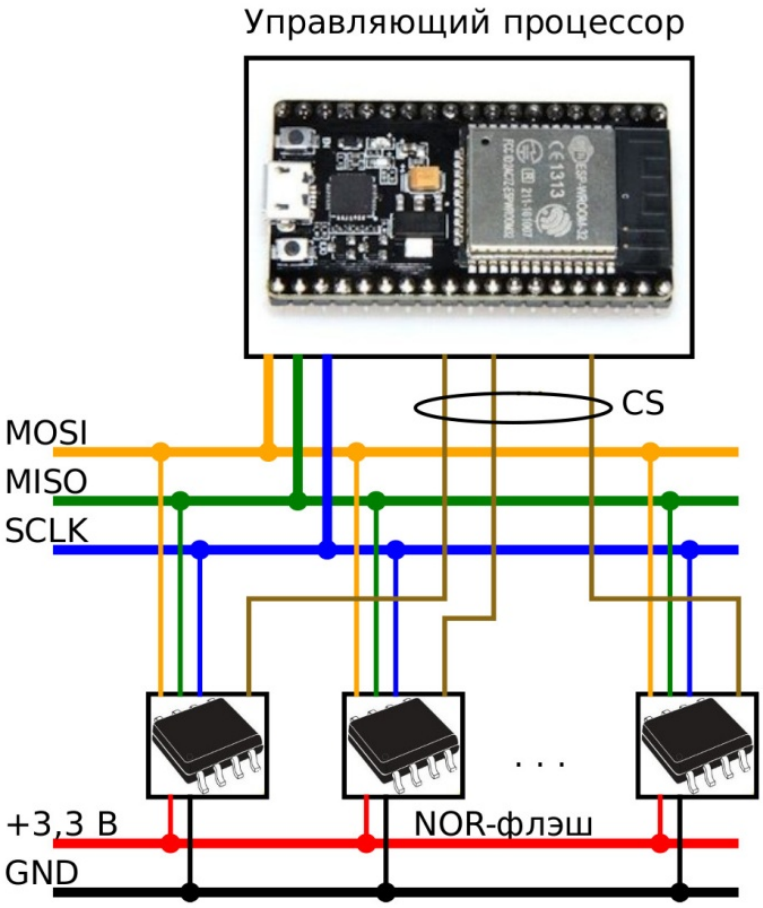

Рис. 3. Разрабатываемая схема системы,
основанной на последовательной отправке команд

Для построения рассмотренной последовательно-параллельной системы предлагается использовать процессорный модуль ESP32, содержащий 2 доступных пользователю свободных интерфейса SPI и 12 свободных портов GPIO, что позволяет подключить до 6 микросхем NOR флеш-памяти к каждому интерфейсу. Использование двухядерного процессора Tensilica Xtensa LX6 позволяет управлять каждым интерфейсом SPI независимо.

Заключение

В ходе проведенной работы авторами получены оценки времени деградирования микросхем флеш-памяти и предложены два варианта построения устройства для принудительной параллельной деградации микросхем NOR-флеш памяти. Предложен вариант выбора оборудования для аппаратной реализации. Рассмотренные системы будут использованы для формирования идентифицирующих микросхем флэш-памяти, предназначенных для проведения статистических исследований.

В дальнейшей работе предполагается расширить возможности устройств, дополнив их пользовательским интерфейсом для управления и сбора статистики, а также провести сравнительное исследование временных характеристик предложенных подходов.

Литература

1. Соколов М. Н., Смолянинова К. А., Якушева Н. А. Проблемы безопасности интернет вещей: обзор // Вопросы кибербезопасности. 2015. № 5 (13). С. 32–35.

2. Бородин А. С., Рожков М. А., Киричек Р. В., Кучерявый А. Е. New IP от 5G к 6G: нужна ли смена парадигмы? // Электросвязь. 2020. № 7. С. 15–21.

3. Wang Y., Yi J., Guo J., Qiao Y., Qi M., Chen Q. A Semistructured Random Identifier Protocol for Anonymous Communication in SDN Network // Security and Communication Networks. 2018. Vol. 2018. p. 2916356. DOI: 10.1155/2018/2916356.

4. Vaidya S., Christensen K. J. A single system image server cluster using duplicated MAC and IP addresses // Proceedings LCN 2001. 26th Annual IEEE Conference on Local Computer Networks, Nov. 2001, Tampa, FL, USA. IEEE, 2002. pp. 206–214. DOI: 10.1109/LCN.2001.990789.

5. Hegde A. MAC Spoofing Detection and Prevention // International Journal of Advanced Research in Computer and Communication Engineering. 2016. Vol. 5. Iss. 1. pp. 229–232. DOI: 10.17148/IJARCCE.2016.5155.

6. Leloglu E. A Review of Security Concerns in Internet of Things // Journal of Computer and Communications. 2017. Iss. 5. pp. 121–136. DOI: 10.4236/jcc.2017.51010.

7. Rao S. P., Holtmanns S., Oliver I., Aura T. Unblocking stolen mobile devices using SS7-MAP vulnerabilities: Exploiting the relationship between IMEI and IMSI for EIR access // 2015 IEEE Trustcom/BigDataSE/ISPA, Aug. 2015, Helsinki, Finland. IEEE, 2015. pp. 1171–1176. DOI: 10.1109/Trustcom.2015.500.

8. Gepko I. General requirements and security architecture for mobile phone anti-cloning measures // IEEE EUROCON 2015 - International Conference on Computer as a Tool (EUROCON), Sep. 2015, Salamanca, Spain. IEEE, 2015. pp. 1–6. DOI: 10.1109/EUROCON.2015.7313666.

9. DeJean G, Kirovski D. RF-DNA: Radio-Frequency Certificates of Authenticity // Lecture Notes in Computer Science. 2007. Vol. 4727. pp. 346–363. DOI: 10.1007/978-3-540-74735-2_24.

10. Wang Y., Yu W., Wu S., Malysa G., Suh G.E., Kan E. C. Flash Memory for Ubiquitous Hardware Security Functions: True Random Number Generation and Device Fingerprints // Proceedings of the 2012 IEEE Symposium on Security and Privacy. 2012. pp. 33−47. DOI: 10.1109/SP.2012.12.

11. Jia S., Xia L., Wang Z., Lin J., Zhang G., Ji Y. Extracting Robust Keys from NAND Flash Physical Unclonable Functions // Lecture Notes in Computer Science. 2015. Vol. 9290. pp. 437−454. DOI: 10.1007/978-3-319-23318-5_24.

12. Jakobsson M., Johansson K.-A. Unspoofable Device Identity Using NAND Flash Memory [Electronic resource] // SecurityWeek: [site]. URL: http://www.securityweek.com/unspoofable-device-identity-using-nand-flash-memory (Accessed date: 18.11.2020).

13. Владимиров С. С., Киричёк Р. В. Методика идентификации устройств интернета вещей на основе принудительной деградации участка флеш-памяти // Электросвязь. 2017. № 2. С. 32−35.

14. Vladimirov S., Kirichek R. The IoT Identification Procedure Based on the Degraded Flash Memory Sector // Lecture Notes in Computer Science. 2017. Vol. 10531. pp. 66−74. DOI: 10.1007/978-3-319-67380-6_6.

15. Vladimirov S. S., Pirmagomedov R., Kirichek R., Koucheryavy A. Unique Degradation of Flash Memory as an Identifier of ICT Device // IEEE Access. 2019. Vol. 7. pp. 107626–107634. DOI: 10.1109/ACCESS.2019.2932804.

16. Владимиров С. С., Янковский А. К. Программное обеспечение для работы с микросхемами памяти с SPI-интерфейсом при исследовании идентификации сетевых устройств // Информационные технологии и телекоммуникации. 2017. Том 5. № 3. С. 74–83.

17. Владимиров С. С., Берестовой Д. М. Протокол идентификации устройств Интернета вещей методом деградированной флэш-памяти // Информационные технологии и телекоммуникации. 2020. Том 8. № 2. С. 20–31. DOI 10.31854/2307-1303-2020-8-2-20-31.

References

1. Sokolov M. N., Smolyaninova K. A., Yakusheva N. A. Problemy bezopasnosti internet-veshchej: obzor // Voprosy kiberbezopasnosti. 2015. No 5. pp. 32–35.

2. Borodin A. S., Rozhkov M. A., Kirichek R. V., Koucheryavy A. E. New IP, from 5G to 6G: do we need a paradigm change? // Electrosvyaz'. 2020. No 7. pp. 15–21.

3. Wang Y., Yi J., Guo J., Qiao Y., Qi M., Chen Q. A Semistructured Random Identifier Protocol for Anonymous Communication in SDN Network // Security and Communication Networks. 2018. T. 2018. p. 2916356. DOI: 10.1155/2018/2916356.

4. Vaidya S., Christensen K. J. A single system image server cluster using duplicated MAC and IP addresses // Proceedings LCN 2001. 26th Annual IEEE Conference on Local Computer Networks, Nov. 2001, Tampa, FL, USA. IEEE, 2002. pp. 206–214. DOI: 10.1109/LCN.2001.990789.

5. Hegde A. MAC Spoofing Detection and Prevention // International Journal of Advanced Research in Computer and Communication Engineering. 2016. Vol. 5. Iss. 1. pp. 229–232. DOI: 10.17148/IJARCCE.2016.5155.

6. Leloglu E. A Review of Security Concerns in Internet of Things // Journal of Computer and Communications. 2017. Iss. 5. pp. 121–136. DOI: 10.4236/jcc.2017.51010.

7. Rao S. P., Holtmanns S., Oliver I., Aura T. Unblocking stolen mobile devices using SS7-MAP vulnerabilities: Exploiting the relationship between IMEI and IMSI for EIR access // 2015 IEEE Trustcom/BigDataSE/ISPA, Aug. 2015, Helsinki, Finland. IEEE, 2015. pp. 1171–1176. DOI: 10.1109/Trustcom.2015.500.

8. Gepko I. General requirements and security architecture for mobile phone anti-cloning measures // IEEE EUROCON 2015 – International Conference on Computer as a Tool (EUROCON), Sep. 2015, Salamanca, Spain. IEEE, 2015. pp. 1–6. DOI: 10.1109/EUROCON.2015.7313666.

9. DeJean G, Kirovski D. RF-DNA: Radio-Frequency Certificates of Authenticity // Lecture Notes in Computer Science. 2007. Vol. 4727. pp. 346–363. DOI: 10.1007/978-3-540-74735-2_24.

10. Wang Y., Yu W., Wu S., Malysa G., Suh G. E., Kan E. C. Flash Memory for Ubiquitous Hardware Security Functions: True Random Number Generation and Device Fingerprints // Proceedings of the 2012 IEEE Symposium on Security and Privacy. 2012. pp. 33–47. DOI: 10.1109/SP.2012.12.

11. Jia S., Xia L., Wang Z., Lin J., Zhang G., Ji Y. Extracting Robust Keys from NAND Flash Physical Unclonable Functions // Lecture Notes in Computer Science. 2015. Vol. 9290. pp. 437–454. DOI: 10.1007/978-3-319-23318-5_24.

12. Jakobsson M., Johansson K.-A. Unspoofable Device Identity Using NAND Flash Memory [Electronic resource] // SecurityWeek: [site]. URL: http://www.securityweek.com/unspoofable-device-identity-using-nand-flash-memory (Accessed date: 18.11.2020).

13. Vladimirov S., Kirichek R. The IoT Devices Identification Procedure based on Forced Degrading of Flash-Memory Sector // Electrosvyaz'. 2017. No. 2. pp. 32–35.

14. Vladimirov S., Kirichek R. The IoT Identification Procedure Based on the Degraded Flash Memory Sector // Lecture Notes in Computer Science. 2017. Vol. 10531. pp. 66–74. DOI: 10.1007/978-3-319-67380-6_6.

15. Vladimirov S. S., Pirmagomedov R., Kirichek R., Koucheryavy A. Unique Degradation of Flash Memory as an Identifier of ICT Device // IEEE Access. 2019. T. 7. pp. 107626–107634. DOI: 10.1109/ACCESS.2019.2932804.

16. Vladimirov S., Yankovskiy A. The Software for Working with SPI Flash Memory Chips in the Study of Network Devices Identification // Telecom IT. 2017. Vol. 5. Iss. 3. pp. 74–83 (in Russian).

17. Vladimirov S., Berestovoy D.: IoT Device Identification Protocol based on Degraded Flash Memory // Telecom IT. 2020. Vol. 8. Iss. 2. pp. 20–31 (in Russian). DOI: 10.31854/2307-1303-2020-8-2-20-31.

Владимиров Сергей Сергеевич – кандидат технических наук, доцент кафедры Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича, vladimirov.opds@gmail.com

Vladimirov Sergey – Candidate of Engineering Sciences, assistant professor, The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, vladimirov.opds@gmail.com

Янковский Антон Константинович – инженер Санкт-Петербургского государственного университета телекоммуникаций им. проф. М. А. Бонч-Бруевича, ostnipx@gmail.com

Yankovskiy Anton – engineer, The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, ostnipx@gmail.com