ВЕСТНИК
ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
МОРСКОГО И РЕЧНОГО ФЛОТА ИМЕНИ АДМИРАЛА С. О. МАКАРОВА

DOI: 10.21821/2309-5180-2017-9-5-1123-1130

OPTIMIZATION OF SECURE DATA TRANSMISSION IN NETWORKS OF AUTOMATED INFORMATION SYSTEMS ON WATER TRANSPORT

S. A. Morozov

Admiral Makarov State University of Maritime and Inland Shipping,
St. Petersburg, Russian Federation

In the article the method of optimization of the work of software tools of automated information systems is considered, by optimizing the operation of the most widespread protocol of secure data transmission, which is used in computer networks. The principles of operation of the ChaCha20 algorithm are described and the possibility of efficient replacement of AES or RC4 with ChaCha20 on platforms that do not have hardware encryption is considered. Such systems are characterized by low computing power. The lack of a hardware encryption mechanism makes the encryption process itself a bottleneck in terms of performance of the entire process of transferring data over secure channels. ChaCha20 is a streaming encryption algorithm from the Salsa20 family. The main operations in the algorithm are the operations quarter round, column round, diagonal round. The input data for the algorithm is the text in clear form. At the output, the algorithm generates a stream of random numbers, with which it makes encryption of the source text. ChaCha20 is a streaming encryption algorithm for the Salsa20 family. The main operations in the algorithm are the operations quarter round, column round, diagonal round. The tests of the performance of the ChaCha20 encryption algorithm along with the cryptographic authentication code (MAC) Poly1305 in TLS v1.2 and TLS v1.3, created by the HTTP / 2, HTTPS clients on different platforms are also demonstrated. The result is the conclusion that the use of the ChaCha20 algorithm as the main encryption algorithm in embedded systems and mobile solutions in encryption algorithms such as TLS / SSL is more preferable in terms of performance than any of the AES variants in including those used in automated control systems for maritime and river transport.

Keywords: transport safety, information security, TLS protocol, ChaCha20 algorithm.

For citation:
Morozov, Sergey A. "Optimization of secure data transmission in networks of automated information systems on water transport." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 9.5 (2017): 1123–1130. DOI: 10.21821/2309-5180-2017-9-5-1123-1130.

УДК 004.421.5

ОПТИМИЗАЦИЯ ЗАЩИЩЁННОЙ ПЕРЕДАЧИ ДАННЫХ В СЕТЯХ АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ НА ВОДНОМ ТРАНСПОРТЕ

С. А. Морозов

ФГБОУ ВО «ГУМРФ имени адмирала С. О. Макарова»,
Санкт-Петербург, Российская Федерация

Рассматривается метод оптимизации работы программных средств автоматизированных информационных систем на морском и речном транспорте путем оптимизации работы самого распространённого протокола защищенной передачи данных, который используется в компьютерных сетях. Описываются принципы работы алгоритма ChaCha20 и рассматривается возможность эффективной замены AES или RC4 на ChaCha20 на платформах, не имеющих аппаратного шифрования. В подавляющем большинстве подобные системы характеризуются низкими вычислительными мощностями. Отсутствие аппаратного механизма шифрования делает сам процесс шифрования «узким местом» в плане производительности всего процесса передачи данных по защищенным каналам. ChaCha20 — это потоковый алгоритм шифрования семейства Salsa20. Основными операциями в алгоритме являются операции quarter round, column round, diagonal round. Входными данными для алгоритма является текст в открытом виде. На выходе алгоритм генерирует поток случайных чисел, с помощью которых

производит шифрование исходного текста. Приводятся результаты тестов производительности алгоритма шифрования ChaCha20 вместе с криптографическим кодом аутентификации (MAC) Poly1305 в TLS v1.2 и TLS v1.3, созданных клиентами HTTP/2, HTTPS на различных платформах. Результатом является вывод о том, что использование алгоритма ChaCha20 как основного алгоритма шифрования во встроенных системах и мобильных решений в алгоритмах шифрованной передачи данных, таких как TLS/SSL, более предпочтительно с точки зрения производительности, чем какой-либо из вариантов AES, в том числе используемых в автоматизированных системах управления на морском и речном транспорте.

Ключевые слова: транспортная безопасность, информационная безопасность, протокол TLS, алгоритм ChaCha20.

Для цитирования:
Морозов С. А. Оптимизация защищённой передачи данных в сетях автоматизированных информационных систем на водном транспорте / С. А. Морозов // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2017. — Т. 9. — № 5. — С. 1123–1130. DOI: 10.21821/2309-5180-2017-9-5-1123-1130.

Введение

В соответствии с Федеральным законом от 09.02.2007 г. № 16-ФЗ «О транспортной безопасности», целями обеспечения транспортной безопасности являются устойчивое и безопасное функционирование транспортного комплекса, а также защита интересов личности, общества и государства в сфере транспортного комплекса от актов незаконного вмешательства. Одним из критериев транспортной безопасности можно считать возможность защищенной передачи данных между информационными системами на морском и речном транспорте [1]. Анализ литературных источников показывает, что на сегодняшний день самым популярным решением в области криптографии являются семейства алгоритмов AES и RC4 [2] – [4]. Эти алгоритмы реализованы и широко используются во всевозможных вариантах крипточипов, которые обеспечивают высокую производительность шифрования. Однако в системах, не имеющих аппаратного шифрования, алгоритмы AES и RC4 не являются оптимальными. Примерами таких систем можно считать большинство встроенных систем (Embedded Systems), в которых присутствует лишь программная реализация шифровального стека.

В связи с непрерывным ростом сетевого трафика и повсеместным использованием защищенных протоколов связи, таких как TLS, увеличивается необходимость в использовании максимально производительных решений, в том числе в области шифрования, и тем самым стимулируются исследования новых решений, как серверных, так и клиентских. Одним из решений, которые могут повысить производительность шифрования на клиентской и серверной стороне, является семейство алгоритмов шифрования Salsa20 [5] и криптопримитив Poly1305 [6]. Если в большинстве серверных и производительных клиентских платформ существует масса аппаратных решений для шифрования данных, то во встроенных системах в большинстве случаев имеет место лишь программная реализация шифровального стека и производительность для них имеет особую актуальность.

Целью данной работы является тестирование одного из алгоритмов семейства Salsa20, а именно ChaCha20, который является наиболее производительным среди Salsa20 [7] на различных платформах, и анализ возможности использования его как основного алгоритма, используемого при передаче информации по шифрованному каналу данных [8].

Методы и материалы

Для того чтобы сформулировать алгоритм ChaCha20, необходимо ввести несколько вспомогательных понятий, а именно quarter round, column round, diagonal round. Основная операция алгоритма ChaCha20 — quarter round (четверть раунда). Она оперирует четырьмя 32-разрядными беззнаковыми целыми числами, обозначим их a, b, c, d. Тогда операция quarter round может быть представлена в *C*-подобной нотации следующим образом:

$$\begin{aligned} a+ &= b;\ d\hat{} = a;\ d <<= 16;\\ c+ &= d;\ b\hat{} = c;\ b <<= 12;\\ a+ &= b;\ b\hat{} = a;\ b <<= 8;\\ c+ &= d;\ b\hat{} = c;\ b <<= 7, \end{aligned} \tag{1}$$

где «+» — целочисленное сложение по модулю 2 ^ 32; «^» — поразрядное исключающее, или (XOR); «<<*n*» — сдвиг влево на *n* бит.

Column round (раунд столбцов). Возьмём шестнадцать 32-разрядных беззнаковых целых чисел и запишем их в виде матрицы 4 × 4:

$$Y_{source} = \begin{pmatrix} y_0 & y_1 & y_2 & y_3 \\ y_4 & y_5 & y_6 & y_7 \\ y_8 & y_9 & y_{10} & y_{11} \\ y_{12} & y_{13} & y_{14} & y_{15} \end{pmatrix}. \tag{2}$$

Отдельно для каждого ряда из матрицы (1) преобразуем слова операцией quarter round:

$$\begin{aligned} (y_0, y_4, y_8, y_{12}) &= quarterround(y_0, y_4, y_8, y_{12});\\ (y_1, y_5, y_9, y_{13}) &= quarterround(y_1, y_5, y_9, y_{13});\\ (y_2, y_6, y_{10}, y_{14}) &= quarterround(y_2, y_6, y_{10}, y_{14});\\ (y_3, y_7, y_{11}, y_{15}) &= quarterround(y_3, y_7, y_{11}, y_{15}). \end{aligned} \tag{3}$$

Diagonal round (раунд диагоналей). Для каждой из четырех диагоналей из матрицы (1) преобразуем слова операцией quarter round:

$$\begin{aligned} (y_0, y_5, y_{10}, y_{15}) &= quarterround(y_0, y_5, y_{10}, y_{15});\\ (y_1, y_6, y_{11}, y_{12}) &= quarterround(y_1, y_6, y_{11}, y_{12});\\ (y_2, y_7, y_8, y_{13}) &= quarterround(y_2, y_7, y_8, y_{13});\\ (y_3, y_4, y_9, y_{14}) &= quarterround(y_3, y_4, y_9, y_{14}). \end{aligned} \tag{4}$$

Формулировка алгоритма ChaCha20

На вход алгоритму подается:

– 256-битный ключ (key);

– 96-битный одноразовый код (nonce);

– 32-битный счетчик операций (block counter);

– текст, который необходимо зашифровать.

Исходная матрица формируется следующим образом:

– первые четыре слова являются константами: 0x61707865, 0x3320646e, 0x79622d32, 0x6b206574;

– 3- – 11-е слово вычитываются из 256-битного ключа, начиная со смещения 0 в порядке байт от младшего к старшему (little-endian order);

– 12-е слово — счетчик операций;

– слова 13 – 15 — одноразовый код, причем 13-е слово является первыми 32-битами nonce'a в порядке байт от младшего к старшему, а 15-е — последними:

$$Y_{source} = \begin{pmatrix} 0x61707865 & 0x3320646e & 0x79622d32 & 0x6b206574 \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ c & n_0 & n_1 & n_2 \end{pmatrix}; \tag{5}$$

k_i=key[i], c=block counter, n_j=nonce[13+j].

Далее над полученной матрицей выполняется 20 раундов, каждый из которых поочередно является либо column round, либо diagonal round, начиная с column round.

Функция, выполняющая раунды

Листинг 1. Функция, выполняющая раунды:

```
DoChaCha20Rounds(4x4matrix y):
  for i = 1 upto 10
    y = columnround(y);
    y = diagonalround(y);
  end
end
```

Далее полученную матрицу складываем по модулю 2 побайтово с исходным текстом. В результате получаем зашифрованный текст с тем же размером в байтах, что и исходный тест.

В качестве примера, возьмем 64-байтный текст и применим к нему алгоритм ChaCha20.

Входные данные:

– 256-битный ключ [*Листинг 2*];

– 96-битный одноразовый ключ [*Листинг 3*];

– счетчик операций: 1;

– исходный текст [*Листинг 4*].

Листинг 2. 256-битный ключ:

```
00000000: 68 57 6d 5a 71 34 74 37 77 21 7a 25 43 2a 46 2d
00000016: 4a 61 4e 64 52 66 55 6a 58 6e 32 72 35 75 38 78
00000032:
```

Листинг 3. 96-битный nonce:

```
00000000: 42 26 45 29 48 40 4d 63 51 66 54 6a
00000012:
```

Листинг 4. Исходный текст:

```
00000000: 54 68 65 20 45 75 72 6f 70 65 61 6e 20 6c 61 6e  | The.European lan
00000016: 67 75 61 67 65 73 20 61 72 65 20 6d 65 6d 62 65  | guages are membe
00000032: 72 73 20 6f 66 20 74 68 65 20 73 61 6d 65 20 66  | rs of the same f
00000048: 61 6d 69 6c 79 2e 20 49 73 20 61 20 6d 79 74 68  | amily.Is a myth
00000064:
```

Исходная матрица:

$$Y_{source} = \begin{pmatrix} 0x61707865 & 0x3320646e & 0x79622d32 & 0x6b206574 \\ 0x03020100 & 0x07060504 & 0x0b0a0908 & 0x0f0e0d0c \\ 0x13121110 & 0x17161514 & 0x1b1a1918 & 0x1f1e1d1c \\ 0x00000001 & 0x00000000 & 0x4a000000 & 0x00000000 \end{pmatrix} \quad (6)$$

После 20 раундов получим следующую матрицу:

$$Y_{result} = \begin{pmatrix} 0x224f51f3 & 0x401bd9e1 & 0x2fde276f & 0xb8631ded \\ 0x8c131f82 & 0x3d2c06e2 & 0x7e4fcaec & 0x9ef3cf78 \\ 0x8a3b0aa3 & 0x72600a92 & 0xb57974cd & 0xed2b9334 \\ 0x794cba40 & 0xc63e34cd & 0xea212c4c & 0xf07d41b7 \end{pmatrix} \quad (7)$$

Для удобства переведем матрицу в вектор в байтах.

Листинг 5. Результирующий вектор в байтах:

```
00000000: 22 4f 51 f3 40 1b d9 e1 2f de 27 6f b8 63 1d ed
00000016: 8c 13 1f 82 3d 2c 06 e2 7e 4f ca ec 9e f3 cf 78
00000032: 8a 3b 0a a3 72 60 0a 92 b5 79 74 cd ed 2b 93 34
00000048: 79 4c ba 40 c6 3e 34 cd ea 21 2c 4c f0 7d 41 b7
00000064:
```

Итоговый текст, полученный в результате побитовой операции XOR результирующего вектора в байтах и исходного текста:

Листинг 6. Итоговый зашифрованный текст:

```
00000000: 6e 2e 35 9a 25 68 f9 80 41 ba 07 28 dd 0d 69 81 | n.5.%h..A..(..i.
00000016: e9 7e 7a ec 1d 43 60 c2 0a 27 af cc fd 9f ae 0b  | .~z..C`..'……
00000032: f9 1b 65 c5 52 47 33 ab 8f 59 3d ab cd 62 b3 57  | ..e.RG3..Y=..b.W
00000048: 16 39 d6 24 e6 51 52 ab 8f 53 0c 35 9f 08 61 d8  | .9.$.QR..S.5..a.
00000064:
```

Результаты

В тестировании производительности будем использовать:

1) пять протоколов шифрования:
   – ChaCha20;
   – AES-128-GCM;
   – AES-256-GCM;
   – AES-128-CBC;
   – AES-256-CBC;
2) два протокола защищенного обмена данными:
   – TLS v1.2;
   – TLS v1.3;
3) два протокола прикладного уровня, для которых будут инициироваться защищенные каналы данных:
   – HTTP/2;
   – HTTPS;
4) шесть платформ, на которых будет производиться тестирование:
   – Intel i7-6700;
   – Intel Xeon E5-2630;
   – Intel ATOM D525;
   – Intel i7-950;
   – AMD FX 8350;
   – AMD Phenom 965.

Для непосредственно шифрования данных в ходе тестирования будем использовать OpenSSL, сконфигурированный поочерёдно для каждого протокола шифрования и для каждой версии TLS, так как данный выбор соответствует типичной и наиболее популярной конфигурации HTTPS сервера [9] – [12]. На всех четырех платформах будем использовать в качестве операционной системы Linux Gentoo (4.11.3). Причем при тестировании алгоритмов семейства AES будет производиться аппаратное ускорение AES-NL для тех процессоров, для которых это возможно. Критерием оценки производительности будет являться скорость шифрования случайно сгенерированного текста размером в 8192 байта. Усредненные результаты за пять тестовых прогонов указаны в Мбайт/с.

Результаты тестирования

Платформы тестирования	Протоколы шифрования				
	ChaCha20	AES-128-GCM	AES-256-GCM	AES-128-CBC	AES-256-CBC
Intel i7-6700*	567	2598	2233	1529	1161
Intel i7-950	421	228	199	319	245
Intel Xeon E5-2630*	256	947	853	538	388
Intel ATOM D525	102	47	41	26	19
AMD FX 8350*	377	1435	1287	721	509
AMD Phenom 965	409	80	62	279	191

* — процессоры, поддерживающие аппаратное ускорение AES-NI.

Как видно из результатов эксперимента, приведенных в таблице, алгоритм AES-128-GCM на платформах с аппаратным ускорением шифрования показал большую производительность по сравнению с ChaCha20 в среднем на 302,91 %, AES-256-GCM — на 256,13 %, AES-128-CBC — на 123,68 %, AES-256-CBC — на 63,77 %, но в то же время на платформах с чисто программным движком шифрования производительность ChaCha20 выше, чем AES-128-GCM, в среднем на 204,3 %, AES-256-GCM — на 279,3 %, AES-128-CBC — на 63,75 %, AES-256-CBC — на 76,21 %.

Заключение

В подавляющем большинстве встроенные системы характеризуются низкими вычислительными мощностями. Отсутствие аппаратного механизма шифрования делает сам процесс шифрования узким местом в плане производительности (Bottle Neck) всего процесса передачи данных по защищенным каналам [13] – [15]. Как видно из эксперимента, алгоритм ChaCha20 оказался более производителен, чем AES-аналоги в системах, не имеющих аппаратного ускорения, например в большинстве чипов, использующихся во встроенных системах, но в то же время уступает им в полномасштабных процессорах, например в серверных решениях, имеющих в своем составе технологии, подобные AES-NI. «Минусами» проведенного эксперимента можно считать недостаточное количество платформ, которые принимали участие в тестировании.

Из ранее изложенного можно сделать вывод о том, что использование алгоритма ChaCha20 как основного алгоритма шифрования во встроенных системах и мобильных решений в алгоритмах шифрованной передачи данных, таких как TLS/SSL, является наиболее предпочтительным с точки зрения производительности, какого-либо иного из вариантов AES, в том числе используемых в автоматизированных системах управления на морском и речном транспорте [16].

СПИСОК ЛИТЕРАТУРЫ

1. *Nyrkov A. P.* Providing the integrity and availability in the process of data transfer in the electronic documents management systems of transport-logistical clusters / A. P. Nyrkov, S. S. Sokolov, S. G. Chernyi, A. V. Chernyakov, A. S. Karpina // Industrial Engineering, Applications and Manufacturing (ICIEAM), International Conference on. — IEEE, 2016. — Pp. 1–4. DOI: 10.1109/ICIEAM.2016.7910915.

2. *Баричев С. Г.* Стандарт AES. Алгоритм Rijdael / С. Г. Баричев, В. В. Гончаров, Р. Е. Серов. — М.: Диалог-МИФИ, 2011. — 176 с.

3. *Biryukov A.* Related-key Cryptanalysis of the Full AES-192 and AES-256 / A. Biryukov, D. Khovratovich // Advances in Cryptology – ASIACRYPT 2009. — Springer Berlin / Heidelberg, 2009. — 412 p. DOI: 10.1007/978-3-642-10366-7.

4. *Daemen J.* AES Proposal: Rijndael / J. Daemen, V. Rijmen // National Institute of Standards and Technology, 2003 — 365 p. — DOI: 10.1109/.2001.962837.

5. *Bernstein D. J.* The Salsa20 family of stream ciphers / D. J. Bernstein // Lecture Notes in Computer Science. — 2008. — Vol. 4986. — Pp. 84–97. DOI: 10.1007/978-3-540-68351-3_8.

6. *Bernstein D. J.* The Poly1305-AES Message-Authentication Code // FSE. — 2005. — Vol. 3557. — Pp. 32–49. DOI: 10.1007/11502760_3.

7. *Bernstein D. J.* ChaCha, a variant of Salsa20 [Электронный ресурс] / D. J. Bernstein. — Режим доступа: https://cr.yp.to/chacha/chacha-20080120.pdf (дата обращения: 10.09.2017).

8. *Aumasson J. P.* New features of Latin dances: analysis of Salsa, ChaCha, and Rumba. / J. P. Aumasson, S. Fischer, S. Khazaei, W. Meier, C. Rechberger // Lecture Notes in Computer Science. — 2008. — Vol. 5086. — Pp. 470–488. DOI: 10.1007/978-3-540-71039-4_30.

9. Dierks T. The Transport Layer Security (TLS) Protocol Version 1.2 / T. Dierks, E. Rescorla. — 2006. — 104 p. DOI: 10.17487/RFC5246.

10. *Polk T.* Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations / T. Polk, K. McKay, S. Chokhani // NIST special publication. — 2014. — Vol. 800. — № 52. — Pp. 32. DOI: 10.6028/NIST.SP.800-52r1.

11. Adobe Systems Incorporated. Enabling HTTP Over SSL [Электронный ресурс]. — Режим доступа: https://docs.adobe.com/docs/en/cq/5-6-1/deploying/config-ssl.html (дата обращения: 10.09.2017).

12. *Agren M.* On Some Symmetric Lightweight Cryptographic Designs / M. Agren. — Belgium, 2012. — 212 p.

13. *Каторин Ю. Ф.* Защищенность информации в каналах передачи данных в береговых сетях автоматизированной идентификационной системы / Ю. Ф. Каторин, В. В. Коротков, А. П. Нырков // Журнал Университета водных коммуникаций. — 2012. — № 1. — С. 98–102.

14. *Nyrkov A. P.* Algorithmic Support of Optimization of Multicast Data Transmission in Networks with Dynamic Routing / A. P. Nyrkov, A. S. Belousov, S. S. Sokolov // Modern Applied Science. — 2015. — Vol. 9. — No. 5. — Pp. 162. DOI: 10.5539/mas.v9n5p162.

15. *Нырков А. П.* Мультисервисная сеть транспортной отрасли / А. П. Нырков, С. С. Соколов, А. С. Белоусов // Вестник компьютерных и информационных технологий. — 2014. — № 4 (118). — С. 33–38. DOI: 10.14489/vkit.2014.04.pp.033-038.

16. *Нырков А. П.* Автоматизированное управление транспортными системами / А. П. Нырков, С. С. Соколов, А. А. Шнуренко. — СПб.: Изд-во ГУМРФ им. адм. С. О. Макарова, 2013. — 325 с.

REFERENCES

1. Nyrkov, A., S. Sokolov, S. Chernyi, A. Chernyakov, and A. Karpina. “Providing the integrity and availability in the process of data transfer in the electronic documents management systems of transport-logistical clusters.” *Industrial Engineering, Applications and Manufacturing (ICIEAM), International Conference on*. IEEE, 2016. DOI: 10.1109/ICIEAM.2016.7910915.

2. Barichev, S. G., V. V. Goncharov, and R. E. Serov. *Standart AES. Algoritm Rijdael*. M.: Dialog-MIFI, 2011.

3. Biryukov, A., and D. Khovratovich. “Related-key Cryptanalysis of the Full AES-192 and AES-256.” *Advances in Cryptology – ASIACRYPT 2009*. Springer Berlin / Heidelberg, 2009. DOI: 10.1007/978-3-642-10366-7.

4. Daemen, J., and V. Rijmen. *AES Proposal: Rijndael*. National Institute of Standards and Technology, 2003. DOI: 10.1109/.2001.962837.

5. Bernstein, Daniel J. “The Salsa20 family of stream ciphers.” *Lecture Notes in Computer Science* 4986 (2008): 84–97. DOI: 10.1007/978-3-540-68351-3_8.

6. Bernstein, Daniel J. “The Poly1305-AES Message-Authentication Code.” *FSE* 3557 (2005): 32–49. DOI: 10.1007/11502760_3.

7. Bernstein, Daniel J. ChaCha, a variant of Salsa20. Web. 10 Sep. 2017 <https://cr.yp.to/chacha/chacha-20080120.pdf>.

8. Aumasson, Jean-Philippe, S. Fischer, S. Khazaei, W. Meier, and C. Rechberger. “New features of Latin dances: analysis of Salsa, ChaCha, and Rumba.” *Lecture Notes in Computer Science* 5086 (2008): 470–488. DOI: 10.1007/978-3-540-71039-4_30.

9. Dierks, T., and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. 2006. DOI: 10.17487/RFC5246.

10. Polk, Tim, Kerry McKay, and Santosh Chokhani. “Guidelines for the selection, configuration, and use of transport layer security (TLS) implementations.” *NIST special publication* 800.52 (2014): 32. DOI: 10.6028/NIST.SP.800-52r1.

11. Adobe Systems Incorporated. Enabling HTTP Over SSL. Web. 10 Sep. 2017 <https://docs.adobe.com/docs/en/cq/5-6-1/deploying/config-ssl.html>

12. Ågren, Martin. *On some symmetric lightweight cryptographic designs*. 2012.

13. Katorin, Ju. F., V. V. Korotkov, and A. P. Nyrkov. “Information security in the channels of data transmission on the waterside networks of automatic identification system.” *Zhurnal Universiteta vodnyh kommunikacij* 1 (2012): 98–102.

14. Nyrkov, Anatoliy Pavlovich, Andrey Sergeevich Belousov, and Sergey Sergeevich Sokolov. “Algorithmic support of optimization of multicast data transmission in networks with dynamic routing.” *Modern Applied Science* 9.5 (2015): 162. DOI:10.5539/mas.v9n5p162.

15. Nyrkov, A. P., S. S. Sokolov, and A. S. Belousov. “Multiservice network of transportation industry.” *Herald of computer and information technologies* 4(118) (2014) 33–38. DOI: 10.14489/vkit.2014.04.pp.033-038.

16. Nyrkov, A. P., S. S. Sokolov, and A. A. Shnurenko. *Avtomatizirovannoe upravlenie transportnymi sistemami*. SPb.: GUMRF imeni admirala S.O. Makarova, 2013.

ИНФОРМАЦИЯ ОБ АВТОРЕ

Морозов Сергей Александрович — аспирант
Научный руководитель:
Нырков Анатолий Павлович —
доктор технических наук, профессор
ФГБОУ ВО «ГУМРФ имени
адмирала С. О. Макарова»
198035, Российская Федерация, Санкт-Петербург,
ул. Двинская, 5/7
e-mail: *s.a.morozzoff@gmail.com*,
kaf _koib@gumrf.ru

INFORMATION ABOUT THE AUTHOR

Morozov, Sergey A. — Postgraduate
Supervisor:
Nyrkov, Anatolij P. —
Dr. of Technical Sciences, professor
Admiral Makarov State University of Maritime
and Inland Shipping
5/7 Dvinskaya Str., St. Petersburg, 198035,
Russian Federation
e-mail: *s.a.morozzoff@gmail.com*,
kaf _koib@gumrf.ru

Статья поступила в редакцию 15 сентября 2017 г.
Received: September 15, 2017.