Научная статья
УДК 004.056.55
ББК 32.973.2+16.84
К 90
DOI: 10.53598/2410-3225-2024-2-341-35-42

Производительность и надежность алгоритмов шифрования в обработке и защите Big Data

(*Рецензирована*)

Альфира Менлигуловна Кумратова[1], Вадим Александрович Залетаев[2], Иван Сергеевич Старчиков[3]

[1–3] *Кубанский государственный аграрный университет имени И. Т. Трубилина*, *Краснодар*, *Россия*
[1] *kumratova.a@edu.kubsau.ru*
[2] *vadimka.zaletaev@mail.ru*
[3] *luck99bahek@mail.ru*

Аннотация. *В представленной статье рассматривается применение различных алгоритмов шифрования в контексте обработки и защиты больших объемов данных. С увеличением объема информации и развитием информационных технологий вопрос обеспечения безопасности данных становится все более актуальным. Проведен сравнительный анализ восьми различных алгоритмов шифрования, включая AES, Blowfish, Camellia, ChaCha20, DES, GOST, RC4, SEED, с использованием языка программирования Java. Статья представляет результаты анализа, включая сравнительные характеристики производительности и надежности каждого из алгоритмов шифрования. Обсуждаются преимущества и недостатки каждого алгоритма, а также делаются выводы о наиболее подходящих алгоритмах для обработки больших объемов данных с точки зрения производительности и надежности.*

Ключевые слова: *алгоритмы шифрования, большие объемы данных, безопасность данных, сравнительный анализ, производительность, надежность, Java, AES, Blowfish, Camellia, ChaCha20, DES, GOST, RC4, SEED*

Для цитирования: *Кумратова А. М., Залетаев В. А., Старчиков И. С. Производительность и надежность алгоритмов шифрования в обработке и защите Big Data // Вестник Адыгейского государственного университета. Сер. : Естественно-математические и технические науки. 2024. Вып. 2 (341). С. 35–42. DOI: 10.53598/2410-3225-2024-2-341-35-42*

Original Research Paper

Performance and reliability of encryption algorithms in processing and protecting Big Data

Alfira M. Kumratova[1], Vadim A. Zaletaev[2], Ivan S. Starchikov[3]

[1–3] *Kuban State Agrarian University named after I. T. Trubilin*, *Krasnodar*, *Russia*
[1] *kumratova.a@edu.kubsau.ru*
[2] *vadimka.zaletaev@mail.ru*
[3] *luck99bahek@mail.ru*

Abstract. *This article examines the application of various encryption algorithms in the context of processing and securing large volumes of data. With the increasing volume of information and the advancement of information technologies, ensuring data security becomes increasingly relevant. A comparative analysis of eight different encryption algorithms, including AES, Blowfish, Camellia, ChaCha20, DES, GOST, RC4, SEED* and *using the Java programming language,* is *carried out. The article presents the results of the analysis, including comparative performance and reliability characteristics of each encryption algorithm. The advantages and disadvantages of each algorithm are discussed, along with conclusions about the most suitable algorithms for processing large volumes of data in terms of performance and reliability are drawn.*

Keywords: *encryption algorithms*, *large volumes of data*, *data security*, *comparative analysis*, *performance*, *reliability*, *Java*, *AES*, *Blowfish*, *Camellia*, *ChaCha20*, *DES*, *GOST*, *RC4*, *SEED*

For citation: *Kumratova A. M., Zaletaev V. A., Starchikov I. S. Performance and reliability of encryption algorithms in processing and protecting Big Data // The Bulletin of the Adyghe State University. Ser. : Natural-Mathematical and Technical Sciences. 2024. Iss. 2 (341). P. 35–42. DOI: 10.53598/2410-3225-2024-2-341-35-42*

С развитием современных информационных систем и переходом общества на повсеместное использование информационных технологий объемы используемых данных, которые необходимо защищать, неуклонно растут. Исходя из этого, потребность в обеспечении безопасности данных является одним из наиболее важных приоритетов для компаний и разработчиков. Шифрование данных – это главный инструмент для обеспечения конфиденциальности и целостности данных.

На сегодняшний момент существует много различных алгоритмов шифрования, но, учитывая ежедневно растущий поток информации, становится необходимым выбор тех алгоритмов, которые способны обеспечить необходимый уровень безопасности, а также эффективно работать с большими объемами данных [1–7].

AES представляет собой симметричный алгоритм шифрования, который повсеместно используется для защиты данных. Алгоритм является стандартным средством шифрования в США и широко применяется во всем мире. Шифр – итеративный блочный, работающий с блоками данных размером 128 бит и ключами размером 128, 192 или 256 бит. Этот алгоритм обеспечивает не только высокий уровень защиты, но и быструю производительность, что показано в таблице 1. Также стоит отметить, что в некоторых чипах предусмотрено аппаратное ускорение для этого алгоритма. Графики зависимости размера от времени для AES и Blowfish представлены на рисунках 1–2.

Таблица 1

Практически полученные значения для алгоритма AES

Table 1. Practically obtained values for the AES algorithm

Длина зашифрованной информации в байтах	Длина расшифрованной информации в байтах	Время шифрования в мс	Время дешифрования в мс	Время выполнения в целом в мс
172282480	172282464	159	136	295
344564944	344564928	280	272	552
516847408	516847392	381	396	778
689129872	689129856	484	506	991
861412336	861412320	606	635	1241
1033694800	1033694784	721	760	1482

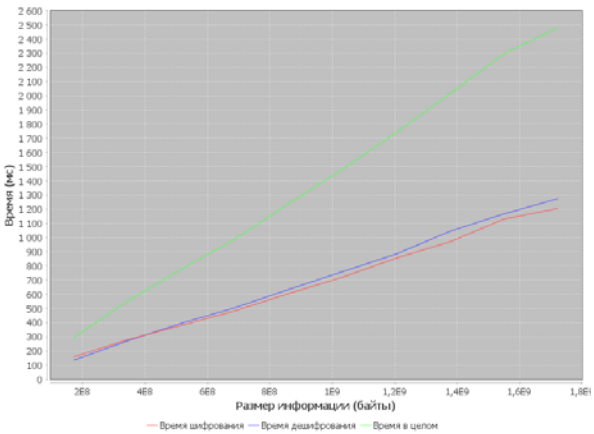

Рис. 1. График зависимости размера от времени для AES

Fig. 1. Size versus time plot for AES

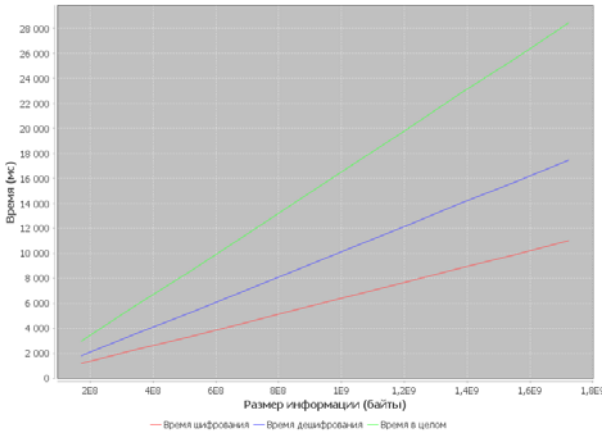

Рис. 2. Графическое представление зависимости размера от времени для Blowfish

Fig. 2. Graphical representation of size versus time for Blowfish

Blowfish является симметричным алгоритмом блочного шифрования, созданным Брюсом Шнайером в 1993 г. На тот момент времени популярным алгоритмом являлся DES, но он уже не соответствовал требованиям времени, поэтому алгоритм Blowfish был представлен как альтернатива DES, обладающая высокой стойкостью к криптоанализу [8–10].

На данный момент он существенно уступает AES как по качеству защиты, так и по производительности, что показано в таблице 2. Также существуют уязвимости при использовании коротких ключей. Тем не менее его следует включить в анализ, поскольку он широко использовался и до сих пор используется. Графики зависимости размера от времени для Camellia и ChaCha20 представлены на рисунках 3–4.

Таблица 2

Практически полученные значения для алгоритма Blowfish
Table 2. Practically obtained values for the Blowfish algorithm

Длина зашифрованной информации в байтах	Длина расшифрованной информации в байтах	Время шифрования в мс	Время дешифрования в мс	Время выполнения в целом в мс
172282472	172282464	1174	1796	2970
344564936	344564928	2272	3545	5817
516847400	516847392	3306	5221	8527
689129864	689129856	4396	6969	11365
861412328	861412320	5520	8708	14228
1033694792	1033694784	6608	10462	17071

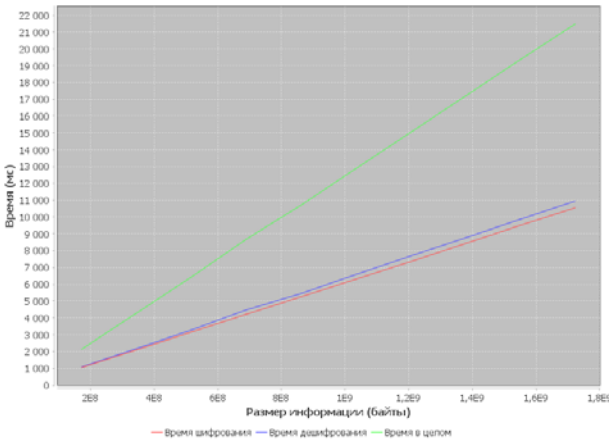

Рис. 3. График зависимости размера от времени для Camellia
Fig. 3. Time-dependent size graph for Camellia

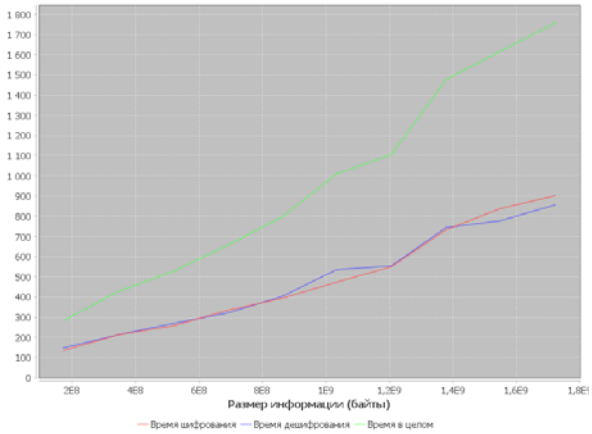

Рис. 4. График зависимости размера от времени для ChaCha20
Fig. 4. Time-dependent size graph for ChaCha20

Camellia представляет собой симметричный алгоритм блочного шифрования. Он разработан японскими и французскими исследователями в криптографии в 2000 г, в качестве замены стандартов AES и DES. Этот блочный шифр имеет переменную длину ключа (128, 192 или 256 бит) и работает с блоками данных размером 128 бит. Этот алгоритм может обеспечить высокий уровень защиты данных, при этом сохраняя относительно хорошую производительность при шифровании больших объемов данных, как показано в таблице 3. В целом можно сказать о нем, как о сбалансированном варианте.

ChaCha20 представляет собой потоковый алгоритм шифрования, идея которого принадлежит Дэниэлу Бернштейну. Он работает с ключами переменной длины (128, 256 или 512 бит) и генерирует псевдослучайные байты на основе инициализационного вектора и счетчика. У него множество преимуществ, включая простоту реализации, высокую скорость работы и высокий уровень безопасности. Этот алгоритм идеально подходит для потокового шифрования. Алгоритм применяется в большом количестве об-

ластей, таких как шифрование коммуникаций в сети, защита данных на устройствах и создание криптографических случайных чисел (таблица 4).

Таблица 3

Практически полученные значения для алгоритма Camellia

Table 3. Practically obtained values for the Camellia algorithm

Длина зашифрованной информации в байтах	Длина расшифрованной информации в байтах	Время шифрования в мс	Время дешифрования в мс	Время выполнения в целом в мс
172282480	172282464	1054	1086	2140
344564944	344564928	2101	2178	4279
516847408	516847392	3153	3276	6429
689129872	689129856	4207	4469	8676
861412336	861412320	5250	5444	10694
1033694800	1033694784	6297	6578	12876

Таблица 4

Практически полученные значения для алгоритма ChaCha20

Table 4. Practically obtained values for the algorithm ChaCha20

Длина зашифрованной информации в байтах	Длина расшифрованной информации в байтах	Время шифрования в мс	Время дешифрования в мс	Время выполнения в целом в мс
172282464	172282464	135	148	283
344564928	344564928	213	213	427
516847392	516847392	257	269	526
689129856	689129856	334	321	655
861412320	861412320	394	403	797
1033694784	1033694784	474	536	1010

DES – это один из старейших алгоритмов блочного шифрования, один из первых действительно качественных алгоритмов, самый популярный в мире алгоритм блочного шифрования. Он разработан в 1970-х гг., является стандартом шифрования в течение длительного времени, но с развитием технологий стало понятно, что он уже не отвечает вызовам современности. Тем не менее, он до сих пор он используется в старых приложениях и протоколах (таблица 5). Графики зависимости размера от времени для DES и GOST представлены на рисунках 5–6.

Таблица 5

Практически полученные значения для алгоритма DES

Table 5. Practically obtained values for the DES algorithm

Длина зашифрованной информации в байтах	Длина расшифрованной информации в байтах	Время шифрования в мс	Время дешифрования в мс	Время выполнения в целом в мс
172282472	172282464	2052	2530	4583
344564936	344564928	4032	5002	9034
516847400	516847392	5981	7462	13443
689129864	689129856	8008	9936	17945
861412328	861412320	9958	12374	22333
1033694792	1033694784	11868	14846	26714

GOST является российским стандартом шифрования, разработанным еще в СССР. Это алгоритм блочного шифрования использует симметричные ключи фиксированной длины в 256 бит (32 байта) и работает с блоками данных размером 64 бита.

Главным его преимуществом остается очень серьезная защита данных, особенно поражает его стойкость к криптоанализу. Но по производительности, вероятно, он не подходит для шифрования больших объемов данных, что показано в таблице 6.

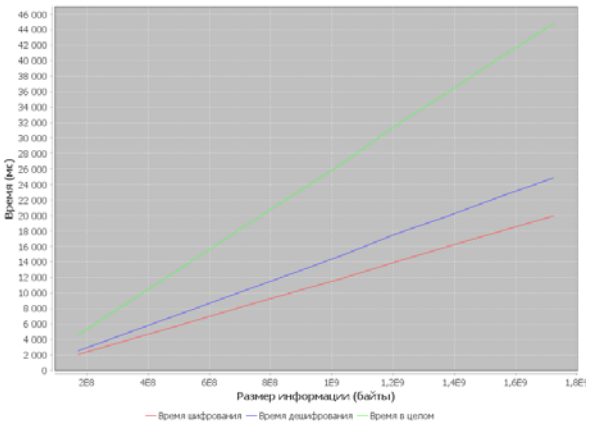

Рис. 5. График зависимости размера от времени для DES

Fig. 5. Time-dependent size graph for DES

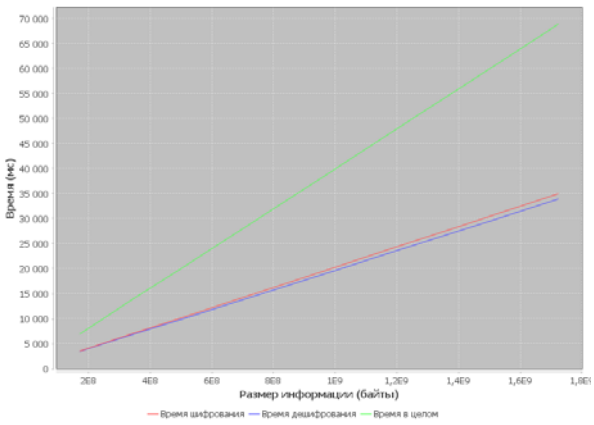

Рис. 6. График зависимости размера от времени для GOST

Fig. 6. Time-dependent size graph for GOS

Таблица 6

Практически полученные значения для алгоритма GOST

Table 6. Practically obtained values for the GOST algorithm

Длина зашифрованной информации в байтах	Длина расшифрованной информации в байтах	Время шифрования в мс	Время дешифрования в мс	Время выполнения в целом в мс
172282472	172282464	3509	3455	6964
344564936	344564928	7060	6887	13948
516847400	516847392	10504	10192	20696
689129864	689129856	14001	13568	27570
861412328	861412320	17483	16941	34424
1033694792	1033694784	20987	20319	41307

RC4 подобно ChaCha20 является алгоритмом потокового шифрования и разработан Рональдом Ривестом в 1987 г. По популярности он стоит в одном ряду с DES и используется в настоящее время в таких протоколах, как SSL/TLS. В его преимущества можно включить простоту реализации и высокую производительность, но, к сожалению, по качеству безопасности он отстал от современности, особенно в сфере защиты от криптоанализа [11–13] (таблица 7). Графики зависимости размера от времени для RC4 и SEED представлены на рисунках 7–8.

Таблица 7

Практически полученные значения для алгоритма RC4

Table 7. Practically obtained values for the RC4 algorithm

Длина зашифрованной информации в байтах	Длина расшифрованной информации в байтах	Время шифрования в мс	Время дешифрования в мс	Время выполнения в целом в мс
172282464	172282464	375	370	746
344564928	344564928	705	705	1411
516847392	516847392	1043	1071	2115
689129856	689129856	1380	1387	2768
861412320	861412320	1734	1741	3475
1033694784	1033694784	2076	2095	4172

SEED – весьма экзотический вариант для наших широт, представляет собой симметричный блочный алгоритм шифрования, родом из Южной Кореи. Он является национальным стандартом в Южной Корее. К главным его характеристикам относится использование 128-битных блоков данных и 128-, 192- или 256-битных ключей. По су-

ти, он представляет собой сеть Фейстеля с 16 раундами шифрования, которые включают в себя различные операции, такие как подстановки и перестановки. Что касается безопасности, она на высоком уровне, но по производительности он уступает AES (таблица 8).

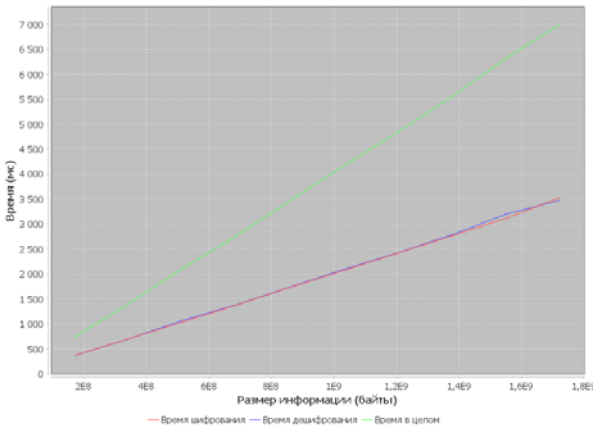

Рис. 7. График зависимости размера от времени для RC4

Fig. 7. Time-dependent size graph for RC4

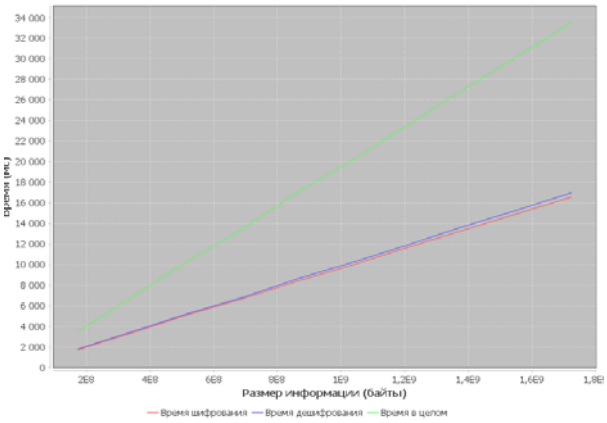

Рис. 8. График зависимости размера от времени для SEED

Fig. 8. A graph of the size versus time for SEED

Таблица 8

Практически полученные значения для алгоритма SEED

Table 8. Practically obtained values for the SEED algorithm

Длина зашифрованной информации в байтах	Длина расшифрованной информации в байтах	Время шифрования в мс	Время дешифрования в мс	Время выполнения в целом в мс
172282480	172282464	1700	1765	3465
344564944	344564928	3372	3494	6866
516847408	516847392	5097	5201	10298
689129872	689129856	6652	6783	13436
861412336	861412320	8379	8600	16979
1033694800	1033694784	9923	10173	20097

В ходе исследования были проанализированы 8 различных алгоритмов. *Целью исследования* является определение алгоритма для использования в IT-проектах. Выбор алгоритма для проекта – это сложный процесс, зависящий от множества факторов. Однако, если обобщить, можно выделить двух явных лидеров как по безопасности, так и по производительности: это ChaCha20 и AES. Сложно сказать, что безопаснее, но можно утверждать, что ChaCha20 работает быстрее и отлично подходит для шифрования больших объемов данных. Важно помнить, что AES не сильно уступает по этим параметрам, при этом он является мировым и рекомендованным стандартом. Авторы считают: если есть возможность – использовать ChaCha20. Все исследования проведены [14] на вычислительной системе с процессором AMD Ryzen 5 5600X 6-core Processor, операционной системой Windows 11 Pro и оперативной памятью объемом 32 ГБ.

Примечания

1. Comparative analysis of different techniques of encryption for secured data transmission / Aquino Valentim Mota, S. Azam, Bharanidharan Shanmugam [et al.] // 2017. IEEE International Conference on Power, Control, Signals & Instrumentation Engineering (ICPCSI). IEEE, 2017. DOI: 10.1109/ICPCSI.2017.8392158

2. Comparative Analysis of Different Encryption Techniques in Mobile Ad-Hoc Networks

(MANETs) // IITM Journal of Management & IT. 2019. Vol. 10, No. 1. P. 55–64.

3. Bhanot R., Rahul H. A review and comparative analysis of various encryption algorithms // International Journal of Security & Its Applications. 2015. No. 9 (4). P. 289–306.

4. Jeeva A. L., Palanisamy Dr V., Kanagaram K. Comparative analysis of performance efficiency and security measures of some encryption algorithms // International Journal of Engineering Research & Applications (IJERA). 2012. No. 2 (3). P. 3033–3037.

5. Comparative analysis of DES, AES, RSA encryption algorithms / Prajapati, Priteshkumar [et al.] // International Journal of Engineering & Management Research (IJEMR). 2014. No. (4) 1. P. 132–6134.

6. Smekal D., HajnyJ., Martinasek Z. Comparative analysis of different implementations of encryption algorithms on FPGA network cards // IFAC-PapersOnLine. 2018. No. 51 (6). P. 312–317.

7. Riman Chadi, Pierre E. Abi-Char. Comparative analysis of block cipher-based encryption algorithms: A survey // Information Security and Computer Fraud. 2015. No. 3 (1). P. 1–7.

8. Comparative analysis of cryptographic algorithms // Marwaha, Mohit [et al.] // Int. J. Adv. Eng. Tech/IV/III/July-Sept. 2013. No. 16. 18 p.

9. Cryptography: a comparative analysis for modern techniques / Maqsood, Faiqa [et al.] / International Journal of Advanced Computer Science & Applications. 2017. No. 8 (6). P. 442–448.

10. Hercigonja Zoran. Comparative analysis of cryptographic algorithms // International Journal of Digital Technology & Economy. 2016. No. 1 (2). P. 127–134.

11. Long Sihan. A Comparative Analysis of the Application of Hashing Encryption Algorithms for MD5, SHA-1, and SHA-512 // Journal of Physics : Conference Series. 2019. Vol. 1314, No. 1. IOP Publishing, 2019. DOI: 10.1088/1742-6596/1314/1/012210

12. Panda Madhumita. Performance analysis of encryption algorithms for security // 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES). IEEE, 2016. DOI: 10.1109/SCOPES.2016.7955835

13. An experimental study on performance evaluation of asymmetric encryption algorithms / Shahzadi Farah, M. Javed, A. Shamim, T. Nawaz // Recent Advances in Information Science, Proceeding of the 3rd European Conf. of Computer Science, (EECS-12). 2012. P. 121–124.

14. VadimKrut/algoritm. URL: https://github.com/VadimKrut/algoritm

References

1. Comparative analysis of different techniques of encryption for secured data transmission / Aquino Valentim Mota, S. Azam, Bharanidharan Shanmugam [et al.] // 2017. IEEE International Conference on Power, Control, Signals & Instrumentation Engineering (ICPCSI). IEEE, 2017. DOI: 10.1109/ICPCSI.2017.8392158

2. Comparative Analysis of Different Encryption Techniques in Mobile Ad-Hoc Networks (MANETs) // IITM Journal of Management & IT. 2019. Vol. 10, No. 1. P. 55–64.

3. Bhanot R., Rahul H. A review and comparative analysis of various encryption algorithms // International Journal of Security & Its Applications. 2015. No. 9 (4). P. 289–306.

4. Jeeva A. L., Palanisamy Dr V., Kanagaram K. Comparative analysis of performance efficiency and security measures of some encryption algorithms // International Journal of Engineering Research & Applications (IJERA). 2012. No. 2 (3). P. 3033–3037.

5. Comparative analysis of DES, AES, RSA encryption algorithms / Prajapati, Priteshkumar [et al.] // International Journal of Engineering & Management Research (IJEMR). 2014. No. (4) 1. P. 132–6134.

6. Smekal D., HajnyJ., Martinasek Z. Comparative analysis of different implementations of encryption algorithms on FPGA network cards // IFAC-PapersOnLine. 2018. No. 51 (6). P. 312–317.

7. Riman Chadi, Pierre E. Abi-Char. Comparative analysis of block cipher-based encryption algorithms: A survey // Information Security and Computer Fraud. 2015. No. 3 (1). P. 1–7.

8. Comparative analysis of cryptographic algorithms // Marwaha, Mohit [et al.] // Int. J. Adv. Eng. Tech/IV/III/July-Sept. 2013. No. 16. 18 p.

9. Cryptography: a comparative analysis for modern techniques / Maqsood, Faiqa [et al.] / International Journal of Advanced Computer Science & Applications. 2017. No. 8 (6). P. 442–448.

10. Hercigonja Zoran. Comparative analysis of cryptographic algorithms // International Journal of Digital Technology & Economy. 2016. No. 1 (2). P. 127–134.

11. Long Sihan. A Comparative Analysis of the Application of Hashing Encryption Algorithms for MD5, SHA-1, and SHA-512 // Journal of Physics : Conference Series. 2019. Vol. 1314, No. 1. IOP Publishing, 2019. DOI: 10.1088/1742-6596/1314/1/012210

12. Panda Madhumita. Performance analysis of encryption algorithms for security // 2016 International Conference on Signal Processing, Communication, Power and Embedded System

(SCOPES). IEEE, 2016. DOI: 10.1109/SCOPES.2016.7955835

13. An experimental study on performance evaluation of asymmetric encryption algorithms / Shahzadi Farah, M. Javed, A. Shamim, T. Nawaz // Recent Advances in Information Science, Proceeding of the 3rd European Conf. of Computer Science, (EECS-12). 2012. P. 121–124.

14. VadimKrut/algoritm**.** URL: https://github.com/VadimKrut/algoritm

Авторы заявляют об отсутствии конфликта интересов.

Статья поступила в редакцию 18.04.2024; *одобрена после рецензирования 28.04.2024*; *принята к публикации 29.04.2024*.

The authors declare no conflicts of interests.

The article was submitted 18.04.2024; *approved after reviewing 28.04.2024*; *accepted for publication 29.04.2024*.