DOI: https://doi.org/10.36910/6775-2524-0560-2020-41-25
УДК: 004.056.55
Журавська Ірина Миколаївна, доктор тех. наук, професор
https://orcid.org/0000-0002-8102-9854
Обухова Катерина Олександрівна, викладач
https://orcid.org/0000-0001-8793-7055
Чорноморський національний університет імені Петра Могили, м. Миколаїв, Україна

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ТА ОЦІНКА КРИПТОСТІЙКОСТІ ШИФРУ RABBIT

Журавська І. М., Обухова К. О. Особливості реалізації та оцінка криптостійкості шифру Rabbit. В роботі розглянуто алгоритм шифру Rabbit, властивості його безпеки та продуктивності. Представлено особливості реалізації шифру Rabbit, який використано для шифрування та дешифрування двійкових даних. Алгоритм Rabbit реалізований на мові С++. Показано, що основною перевагою використання алгоритму Rabbit як потокового шифру є поєднання швидкості та ефективності (рівня безпеки). Досліджено криптостійкість шифру Rabbit під атакою грубої сили з вирахуванням часу перебору ключа (часу секретності інформації) за умови, що алгоритм ідеальний. Доведено, що з урахуванням закону Мура час криптоаналізу скорочується більш ніж на порядок.
Ключові слова: шифрування, потокові шифри, шифр Rabbit, особливості реалізації шифру, оцінка криптостійкості, закон Мура.

Журавская И. Н., Обухова Е. А. Особенности реализации и оценка криптостойкости шифра Rabbit. В работе рассмотрен алгоритм шифра Rabbit, свойства его безопасности и производительности. Представлены особенности реализации шифра Rabbit, который используется для шифрования и дешифрования двоичных данных. Алгоритм Rabbit реализован на языке С++. Показано, что основным преимуществом использования алгоритма Rabbit как потокового шифра является сочетание скорости и эффективности (уровня безопасности). Исследована криптостойкость шифра Rabbit под атакой грубой силы за вычетом времени перебора ключа (времени секретности информации) при условии, что алгоритм идеален. Доказано, что на основе закона Мура время криптоанализа сокращается более чем на порядок.
Ключевые слова: шифрование, потоковые шифры, шифр Rabbit, особенности реализации шифра, оценка криптостойкости, закон Мура.

Zhuravska I. M., Obukhova K. O. Features of implementation and assessment of Rabbit cipher cryptographically strong. The paper considers the Rabbit cipher algorithm, its security and performance properties. The features of the Rabbit cipher implementation, which are used to encrypt and decrypt binary data, are presented. The Rabbit algorithm is realized in C++. It is shown that the main advantage of using the Rabbit algorithm as a stream cipher is the combination of speed and efficiency (security level). The cryptographic strength of the Rabbit cipher under a brute-force attack is investigated, the time to brute force the key (time of information secrecy) is calculated, provided that the algorithm is perfect. It has been proven that, taking into account Moore's law, the cryptanalysis time is reduced by more than an order of magnitude.
Keywords: encryption, stream cipher, Rabbit cipher, cipher implementation, cryptosecurity evaluation, Moore's law.

Постановка проблеми та аналіз досліджень.

Сьогодні, завдяки інтенсивному розвитку та вдосконаленню технологій, суспільство має більше можливостей для соціальної взаємодії, обробки та передачі інформації, яка може служити інтересам окремих людей, корпорацій чи держав. Тому, використання інформації в різних сферах діяльності та життя людини вимагає різноманітних засобів захисту, за умови надання політики конфіденційності послуг в інформаційних та телекомунікаційних системах. Для розв'язання проблем, пов'язаних з безпекою даних, криптографія використовується не тільки для зберігання, а також для передачі даних. Криптографічний алгоритм має дві основні характеристики: можливість захистити дані від різних атак та швидкість обробки. Алгоритм вважається захищеним, якщо не існує атак, здатних розкрити вихідний вміст без знання ключа.

Отже, сила будь-якого захищеного алгоритму шифрування, як правило, вимірюється на основі складності отримання ключа шифрування за допомогою кібератак, таких як груба сила. Передбачається, що чим більше розмір ключа, тим зловмиснику важче обчислити цей ключ. Тому, зі збільшенням розміру ключа зазвичай збільшується обчислювальна складність і час обробки алгоритмів.

В даний час у криптографії існує кілька груп криптографічних перетворень, а саме симетричне та асиметричне шифрування. Криптографічні примітиви симетричного шифрування використовують методи криптографічного перетворення для відкритого тексту та той самий секретний ключ для тексту шифру (наприклад, алгоритми блоку та потоку), тоді як криптографічні примітиви для асиметричного шифрування використовують пару ключів, які взаємопов'язані [1].

Особлива увага приділяється потоковим криптографічним перетворенням, призначеним для захисту інформаційних й телекомунікаційних систем та технологій у більшості криптографічних додатків, включаючи генерацію псевдовипадкових послідовностей, шифрування для інформаційних та

телекомунікаційних систем, а також забезпечення конфіденційності та цілісності інформації для криптографічних протоколів автентифікації, електронних підписів та інших послуг [2].

Реалізація міжнародних проектів, таких як eSTREAM [3], NESSIE, CRYPTREC (в Японії) [4], це підтверджує. Вони спрямовані на розробку та дослідження алгоритмів шифрування, які забезпечували б високий рівень криптографічної стабільності, високу продуктивність та функціонування на різних обчислювальних платформах. Через ці проекти були прийняті національні та міжнародні стандарти криптографічного перетворення [5].

Потоковий шифр Rabbit є одним із шифрів проекту eSTREAM, він був вперше представлений на Fast Software Encryption (FSE) 2003 [6]. Шифр Rabbit характеризується високою продуктивністю в програмному забезпеченні із виміряною швидкістю шифрування/дешифрування.

Метою роботи є дослідження особливостей реалізації та оцінка криптостійкості потокового шифру Rabbit.

Виклад основного матеріалу й обґрунтування отриманих результатів дослідження.

Симетрична криптографія (або симетричне шифрування) – це тип схеми шифрування, в якій один і той же ключ використовується як для шифрування, так і для дешифрування повідомлень. Такий метод кодування інформації в основному застосовувався в останні десятиліття для сприяння таємному спілкуванню між урядами та військовими. В наш час симетричні ключові алгоритми широко застосовуються в різних типах комп'ютерних систем для підвищення безпеки даних [7].

Дві найпоширеніші схеми симетричного шифрування, що використовуються сьогодні, базуються на блокових та потокових шифрах. Блокові шифри групують дані у блоки заздалегідь визначеного розміру, і кожен блок шифрується за допомогою відповідного ключа та алгоритму шифрування (наприклад, 128-бітний відкритий текст шифрується у 128-бітний шифротекст).

Потокові шифри використовують ключ для генерації стійкої псевдовипадкової послідовності довжини повідомлення, яка накладається на відкритий текст за допомогою операції ⊕ XOR. Розшифрування повідомлення проходить так само, як і шифрування.

Шифр Rabbit

Цей алгоритм наряду з алгоритмом Salsa20 є переможцем проекту eStream. Метою проекту був пошук стійких потокових алгоритмів шифрування. Алгоритм **Rabbit** використовує 128-бітний ключ та 64-бітний вектор ініціалізації [8].

У конструкції шифру помітна конструкція wide-pipe, стан шифру складається із 256 бітів внутрішнього стану та 256-бітів лічильника. Бент-функція алгоритму виглядає як квадрат суми оновленого *i*-го значення лічильника та *j*-го значення стану. Результат 5-ти або 9-ти (з установкою вектору ініціалізації) раундів криптографічного перетворення додається за модулем два (див. ⊕ XOR) до відкритого тексту [9].

Переваги та слабкості шифру Rabbit

Із переваг алгоритму слід зазначити, компактний дизайн, надійність та швидкість роботи. Шифр Rabbit був розроблений таким чином, щоб він був швидшим, ніж широко використовувані шифри, і виправдовував розмір ключа 128 біт для шифрування до 264 блоків відкритого тексту. Це означає, що для зловмисника, який не знає ключа, не повинно бути можливості відрізнити до 264 блоків виводу шифру від виводу справді випадкового генератора, використовуючи менше кроків, ніж це потрібно для вичерпного пошуку ключів. З 2003 р. і дотепер немає відомостей про практичні атаки або вразливості шифру Rabbit.

Єдиною слабкістю алгоритму є за сьогоднішніми мірками невелика довжина ключа та ініціалізаційного вектору. На жаль, шифр не має можливості масштабування, і через це його все частіше відносять до алгоритмів легкої криптографії.

Алгоритм Rabbit

1. *Схема установлення ключів*

Першим кроком в алгоритмі є встановлення ключа. 128-бітовий ключ ділиться на вісім підключів, кожен з 16 біт (рис. 1). Потім змінні стану та змінні лічильника обчислюються на основі підключів. Далі система повторюється чотири рази, відповідно до функції наступного стану, щоб зменшити кореляцію між бітами ключа та змінними внутрішнього стану.

2. *Схема установки вектору ініціалізації*

Другий крок – це схема встановлення вектору ініціалізації. Схема налаштування вектору ініціалізації працює, модифікуючи стан лічильника як функцію вектору ініціалізації, що робиться шляхом XOR'у 64-бітного вектору ініціалізації на всіх 256-бітах змінних лічильника.

3. *Функція наступного стану*

Наступним кроком є функція наступного стану. Ця функція – ядро алгоритму Rabbit, що забезпечує правильне поєднання бітів вектору ініціалізації, використовуючи значення лічильників та державних реєстрів.

4. Система лічильників

Наступним кроком є система лічильників, де лічильники реєстрів оновлюються шляхом поєднання поточного стану всіх лічильників регістрів з постійним значенням і значення біта переносу.

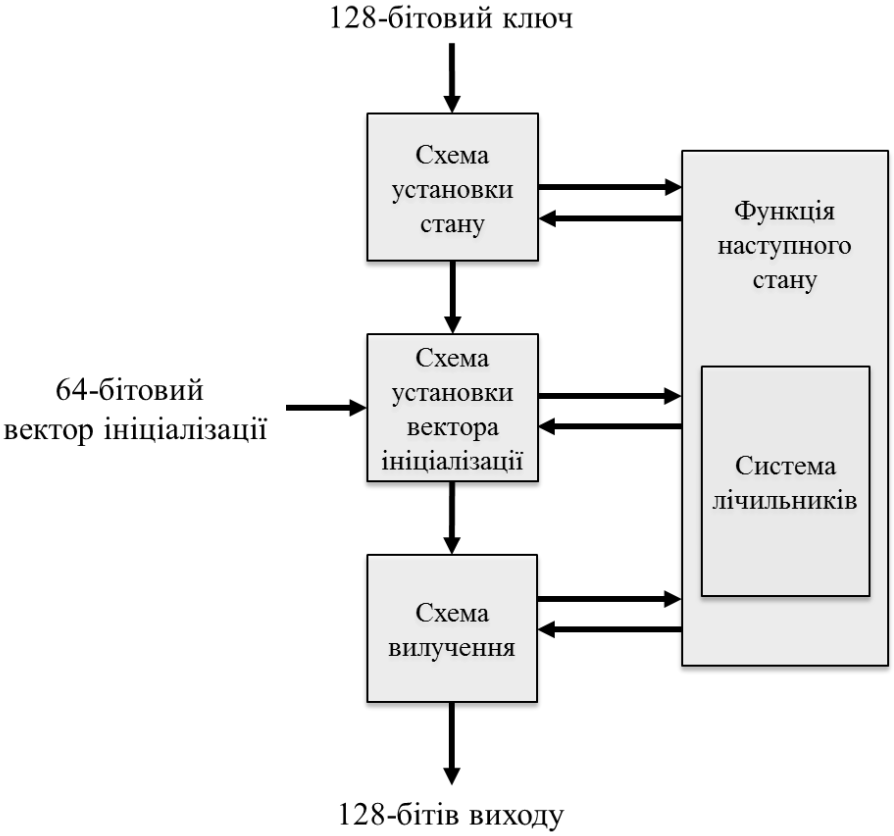

Рис. 1. Структурна схема алгоритму Rabbit

5. Схема вилучення

Останній крок – схема видобутку, у якій операція XOR застосовується до різних регістрів стану для створення восьми 16-бітових регістрів потоків ключів. Потім ці біти ключа використовуються для XOR із потоком бітів простого тексту.

Програмна реалізація шифру Rabbit

Було реалізовано сценарій на основі потокового шифру Rabbit (на C++ з виводом у консолі), що дозволяє шифрувати та розшифровувати двійкові файли.

Слід зазначити, що всі дані вводу у термінал система записує у текстові файли. Наприклад, в UNIX – це .bash_history, у наведеній реалізації – це файл ConsoleHost_history.txt. Тому поле для вводу пароля слід робити прихованим (рис. 2, рис. 3).

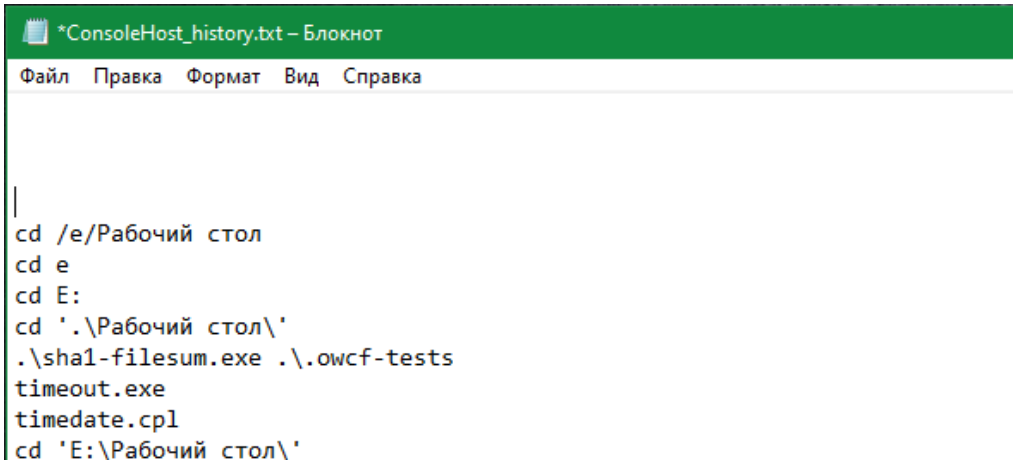

Рис. 2. Історія терміналу з всіма даними вводу (введення пароля приховано)

У даній реалізації, перетворення пароля у ключ шифрування відбувається за допомогою стискаючої функції на основі шифру Rabbit. Насправді для перетворення пароля у надійний ключ слід використовувати алгоритм PBKDF2 та його похідні.

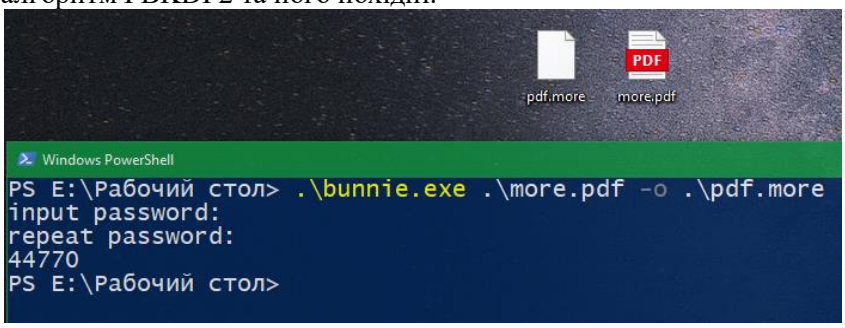

Рис. 3. Результат успішного шифрування файлу *more.pdf* (введення пароля приховано)

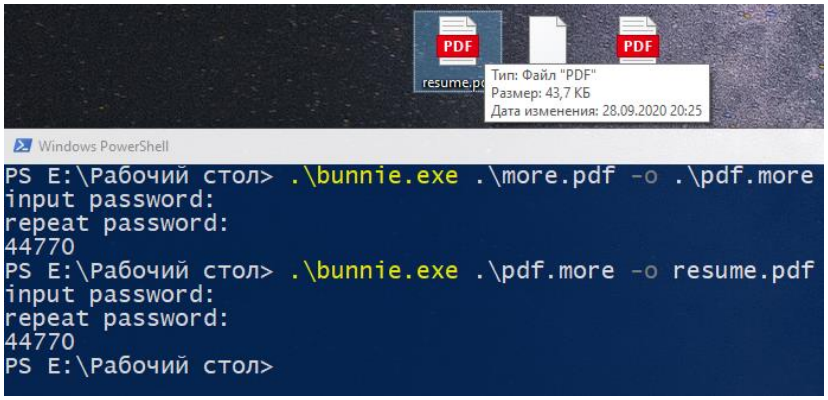

Рис. 5. Результат розшифрування файлу *pdf.more*

Далі проводиться перевірка контрольних сум файлу оригінала (*more.pdf*) та розшифрованого файлу (*resume.pdf*).

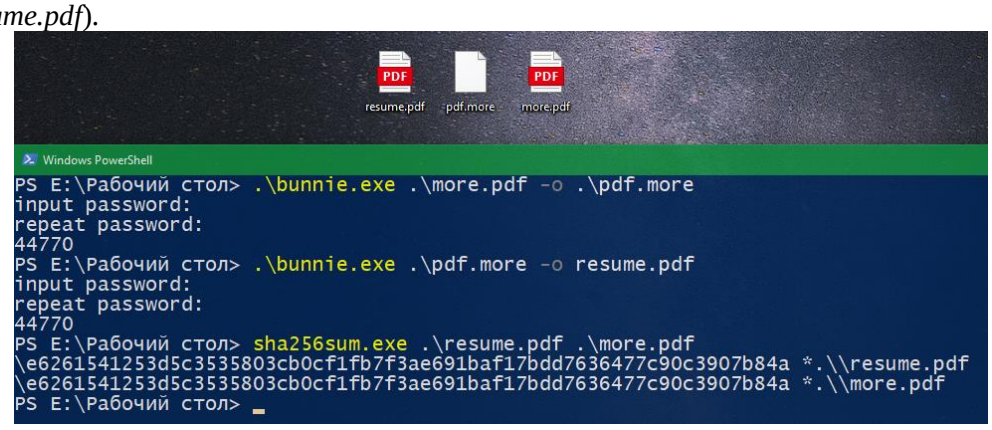

Рис. 6. Звірка контрольних сум оригіналу та розшифрованого файлу

Оскільки при перевірці контрольні суми обох файлів (more.pdf та resume.pdf) збігаються, то можна зробити висновок, що вміст вихідного файлу не був пошкоджений або змінений при перетвореннях.

Оцінка крипостійкості

Щодо крипостійкості шифру Rabbit, то вона вимірюється повним перебором ключа, тобто треба перебрати 2^{128} можливих ключів. Розглянемо повний перебір усіх можливих ключів. Формула для розрахунку буде мати наступний вигляд:

$$T = \frac{S^L}{V*t}, \quad (1)$$

де L – довжина ключа (кількість двійкових розрядів символу алфавіту * кількість символів ключа);
S – потужність алфавіту ключа. Алфавіт системи числення – це множина цифр, використовуваних в ній. Основа системи числення дорівнює потужності алфавіту. Тобто, S=2 (0 або 1).
V – швидкість перебору (операцій на секунду або флопс);
t – коефіцієнт переводу секунд у роки (t=60*60*24*365=31536000);
T – час повного перебору (в роках).

Наприклад, **Core i5-9300H** – процесор для ноутбуків Lenovo Legion y540 (Core i5-9300H 2400 MHz, 8 Cores) складається з восьми обчислювальних ядер.

$$V=F(\text{МГц})*n(\text{ядер})*4*10^6 \quad (2)$$

Розрахуємо кількість виконуваних операцій в секунду для наведеної моделі процесору:

$$V=2400\ \text{МГц}*8*4*10^6=76{,}8\ \text{Гфлопс}$$

Отже, при довжині ключа 64 біти його повний перебір відбудеться за:

$$T=\frac{2^{64}}{76{,}8*10^9}=\frac{500*10^6}{36{,}8*10^9}\approx 7{,}6\ \text{років},$$

Якщо при відомій довжині ключа (у бітах) його повний перебір займає набагато більше ніж час, зазначений у табл.1, тоді істотно, що така інформації вже більше нікому буде не потрібна.

В таблиці 1 (за Шнайєром) наведені дані щодо необхідного часу дотримання режиму секретності інформації різних типів [10].

Таблиця 1. Вимоги до безпеки різної інформації

Типи трафіку	Час життя	Мінімальна довжина ключа (в бітах)
Тактична військова інформація	Хвилини/години	56-64
Оголошення о продуктах, злитті компаній, процентних ставках	Дні/тижні	64
Довготривалі бізнес-плани	Роки	64
Торгові секрети (наприклад, рецепт Coca-Cola)	Десятиліття	112
Секрети водневої бомби	>40 років	128
Особи шпигунів	>50 років	128
Особисті справи	>50 років	128
Дипломатичні конфлікти	>65 років	128

Якщо необхідно дізнатись якійсь секрет (військовий або комерційний), можливо забезпечити значно більшу швидкість. Швидкість перебору залежить й від кількості вкладених грошей.

Таблиця 2. Оцінка середнього часу для вскриття шифру методом грубої сили (повного перебору всіх ключів)

Час крипто аналізу, років	Довжина ключів у бітах					
	40	56	64	80	112	128
Без урахування закону Мура	$4{,}5*10^{-7}$	0,029	7,6	$4{,}9*10^5$	$2{,}1*10^{15}$	$1{,}4*10^{20}$
З урахуванням закону Мура	$2{,}0*10^{-7}$	$1{,}9*10^{-6}$	$2{,}8*10^{-6}$	$4{,}5*10^{-6}$	$8{,}1*10^{-6}$	$9{,}8*10^{-6}$

Відповідно до Закону Мура [11], кожні 5 років потужність обчислювальної техніки збільшується у 10 разів (2 рази на рік).

Тобто, треба перетворити формулу (1) з урахуванням закону Мура. Отримаємо:

$$\sum_1^T x(t) = \frac{S^L}{t}, \quad (3)$$

де

$$x(t) = V*10^{T/5}.$$

Далі, перетворюємо отриману геометричну прогресію (q=$10^{1/5}$), найдемо суму та отримаємо T:

$$\mathrm{T} = 5*\lg\left(\frac{S^L*\left(10^{1/5}-1\right)}{10^{1/5}*V*t}+1\right), \quad (4)$$

Таким чином 50 % ключів знаходяться при переборі половини комбінацій, тоді:

$$T_C = 5*\lg\left(\frac{S^L*\left(10^{1/5}-1\right)}{2*10^{1/5}*V*t}+1\right), \quad (5)$$

Таким чином, час криптоаналізу скорочується більш ніж на порядок:

$$T_C = 5*\lg\left(\frac{2^{64}*\left(10^{1/5}-1\right)}{2*10^{1/5}*76{,}8*10^9}+1\right) \approx 88 \text{ с.}$$

Тим самим отримано формулу за якою можна вирахувати час перебору ключа (час секретності інформації), за умови що алгоритм ідеальний.

Висновки та перспективи подальшого дослідження. Було розглянуто алгоритм шифру Rabbit, властивості його безпеки та продуктивності. Представлено особливості реалізації шифру Rabbit, який використано для шифрування та дешифрування двійкових даних. Алгоритм Rabbit реалізований на мові C++. Основною перевагою використання алгоритму Rabbit як потокового шифру є поєднання швидкості та ефективності (рівня безпеки). Досліджено криптостійкість шифру Rabbit під атакою грубої сили з вирахуванням часу перебору ключа (часу секретності інформації) за умови, що алгоритм ідеальний. Показано, що з урахуванням закону Мура час криптоаналізу скорочується більш ніж на порядок.

Список бібліографічного опису

1. Ferguson N., Schneier B. Practical Cryptography, John Wiley & Sons, 2003, 432p.
2. Menezes A.J., van Oorschot P.C., Vanstone S.A. Handbook of Applied Cryptography, CRC Press, 1997, 794 p.
3. The eSTREAM Project, 2004, URL: http://www.ecrypt.eu.org (Last accessed: 25.11.2020).
4. Cryptography Research and Evaluation Committees, CRYPTREC, 2005. URL: http://www.cryptrec.go.jp (Last accessed: 25.11.2020).
5. Gorbenko I., Kuznetsov A., Gorbenko Y., Vdovenko S., Tymchenko V., Lutsenko M. Studies on statistical analysis and performance evaluation for some stream ciphers. In International Journal of Computing, 18(1) 2019, 82-88
6. The eSTREAM Project eSTREAM Phase 3. Rabbit (Portfolio Profile 1). URL: http://www.ecrypt.eu.org (Last accessed: 25.11.2020).
7. Murtaza A., Pirzada S. J. H., Jianwei L. A New Symmetric Key Encryption Algorithm With Higher Performance. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies*. Sukkur, Pakistan, 30-31 January 2019.
8. Boesgaard M., Vesterager M., Pedersen T., Christiansen J., and Scavenius O. Rabbit: A New High-Performance Stream Cipher. URL: https://web.archive.org/web/20050629021516/http://www.cryptico.com/Files/filer/rabbit_fse.pdf (Last accessed: 25.11.2020).
9. De Canni`ere C., Lano J., Preneel B. Comments on the Rediscovery of Time Memory Data Tradeoffs. URL: https://www.ecrypt.eu.org/stream/papersdir/040.pdf (Last accessed: 25.11.2020).
10. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы и исходные коды на языке С. 2-е изд. Киев : Диалектика, 2016. 1040 с.
11. Половинко І., Кріль Т. Фізика оптичних комп'ютерів. *Електроніка та інформаційні технології.* 2014. Вип. 4. С. 3–23.

References

1. Ferguson N., Schneier B. Practical Cryptography, John Wiley & Sons, 2003, 432p.
2. Menezes A.J., van Oorschot P.C., Vanstone S.A. Handbook of Applied Cryptography, CRC Press, 1997, 794 p.
3. The eSTREAM Project, 2004, URL: http://www.ecrypt.eu.org (Last accessed: 25.11.2020).
4. Cryptography Research and Evaluation Committees, CRYPTREC, 2005. URL: http://www.cryptrec.go.jp (Last accessed: 25.11.2020).
5. Gorbenko I., Kuznetsov A., Gorbenko Y., Vdovenko S., Tymchenko V., Lutsenko M. Studies on statistical analysis and performance evaluation for some stream ciphers. In International Journal of Computing, 18(1) 2019, 82-88
6. The eSTREAM Project eSTREAM Phase 3. Rabbit (Portfolio Profile 1). URL: http://www.ecrypt.eu.org (Last accessed: 25.11.2020).
7. Murtaza A., Pirzada S. J. H., Jianwei L. A New Symmetric Key Encryption Algorithm With Higher Performance. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies*. Sukkur, Pakistan, 30-31 January 2019.
8. Boesgaard M., Vesterager M., Pedersen T., Christiansen J., and Scavenius O. Rabbit: A New High-Performance Stream Cipher. URL: https://web.archive.org/web/20050629021516/http://www.cryptico.com/Files/filer/rabbit_fse.pdf (Last accessed: 25.11.2020).
9. De Canni`ere C., Lano J., Preneel B. Comments on the Rediscovery of Time Memory Data Tradeoffs. URL: https://www.ecrypt.eu.org/stream/papersdir/040.pdf (Last accessed: 25.11.2020).
10. Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. 2nd ed. Kyiv : Dialektika, 2016. 1040 p.
11. Polovynko I., Kril T. The Physics of Optical Computers. *Electronics and information technologies*. 2014. Is. 4. P. 3–23.