ISSN 2519-2310

CS&CS Journal

KARAZIN UNIVERSITY
CLASSICS AHEAD OF TIME

4(16) 2019

COMPUTER SCIENCE AND CYBERSECURITY

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені В.Н.КАРАЗІНА
ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ имени В.Н. КАРАЗИНА
V.N. KARAZIN KHARKIV NATIONAL UNIVERSITY

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА
КОМПЬЮТЕРНЫЕ НАУКИ И КИБЕРБЕЗОПАСНОСТЬ
COMPUTER SCIENCE AND CYBERSECURITY
(CS&CS)

Issue 4(16) 2019

Заснований 2015 року

Міжнародний електронний науково-теоретичний журнал
Международный электронный научно-теоретический журнал
International electronic scientific journal

The journal publishes research articles on theoretical, scientific and technical problems of effective facilities development for computer information communication systems and on information security problems based on advanced mathematical methods, information technologies and technical means.

Journal is published quarterly.
Approved for placement on the Internet by Academic Council of the Karazin Kharkiv National University (April 27, 2020, protocol No.8)

The journal has Digital Object Identifier: **10.26565/2519-2310**.

Editorial office:
V.N. Karazin Kharkiv National University
Svobody Sq., 6, office 315a, Kharkiv, 61022, Ukraine *(North building of University, 3th floor)*
Phone: +38 (057) 705-10-83
E-mail: *cscsjournal@karazin.ua*
Web-page: *http://periodicals.karazin.ua/cscs (Open Journal System)*

Published articles have been internally and externally peer reviewed

TABLE OF CONTENTS

Issue 4(16) 2019

УДК 004.056.5:621.391 DOI: 10.26565/2519-2310-2019-4-01

ПСЕВДОВИПАДКОВІ ДИСКРЕТНІ ПОСЛІДОВНОСТІ ДЛЯ СТЕГАНОСИСТЕМ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ПРЯМОГО РОЗШИРЕННЯ СПЕКТРА

Олексій Смірнов[1], Анна Арищенко[2], Євгеній Деменко[2], Олександр Онікійчук[2], Олександр Кузнецов[2]

[1] - Центральний український НТУ, Кропивницький, Україна
[2] - Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, Харків, 61022, Україна
dr.smirnovoa@gmail.com, annaarischenko@gmail.com, demenjay@gmail.com, onik4524a@gmail.com, kuznetsov@karazin.ua

Рецензент: Микола Карпінський, д.т.н., проф., університет Бельсько-Бяла, Бельсько-Бяла, Польща.
mkarpinski@ath.bielsko.pl

Надійшло: Січень 2020.

Анотація. *В статті розглядаються псевдовипадкові дискретні послідовності (сигнали), які використовуються для стеганографічного приховування інформаційних повідомлень у контейнерах-зображеннях. Для приховування застосовується технологія прямого розширення спектра, суть якої полягає в модуляції інформаційних даних довгими псевдовипадковими (шумовими) послідовностями. Повідомлення набувають вигляду шуму, через що виявити таку передачу вкрай складно. В роботі досліджуються різні способи формування дискретних сигналів і оцінюється інтенсивність помилок при відновленні повідомлень. Виявлено, що спосіб формування псевдовипадкових послідовностей впливає на інтенсивність помилок, тому в роботі обґрунтовується вибір найбільш придатних сигналів. Крім того, оцінюється викривлення контейнера-зображення в результаті приховування даних. Стаття містить переважно результати експериментальних досліджень, які можуть бути корисними при обґрунтуванні різних варіантів побудови стеганографічних систем з прямим розширенням спектра.*

Ключові слова: *приховування інформації; стеганографія; технологія прямого розширення спектру; псевдовипадкова послідовність; розширюючі сигнали.*

1 Вступ

Для приховування даних в контейнерах-зображеннях використовують різні стеганографічні методи [1-4]. Найбільш цікавим підходом є застосування технології прямого розширення спектру [5-17]. Технологія прямого розширення спектра традиційно використовується в системах радіозв'язку з множинним доступом [18-21]. Вона заснована на модуляції інформаційних повідомлень так званими розширюючими сигналами - довгими псевдовипадковими послідовностями, що мають хаотичний, шумоподібний вид. У цьому випадку передане повідомлення стає подібно шуму і його дуже складно розпізнати. Крім того, застосовані методи кореляційного прийому складних шумоподібних сигналів дозволяють виправляти помилки, підвищуючи тим самим завадостійкість системи зв'язку.

У роботах [5-17] технологія прямого розширення спектра частот застосовується для приховування інформаційних повідомлень у цифрових контейнерах-зображеннях. Наприклад, в [5-11] пропонувалося використовувати нелінійну модуляцію псевдовипадковими послідовностями, елементи яких розподілені за нормальним законом з нульовим середнім і одиничним середньоквадратичним відхиленням. Дійсно, інтерпретуючи зображення як шум в каналі зв'язку (КЗ), вдається приховати інформаційні повідомлення при прийнятному рівні внесених викривлень в контейнер.

Ціллю даної статті є дослідження різних варіантів формування розширюючих сигналів, а також їх впливу на якісні характеристики стеганосистеми. Зокрема, в роботі оцінюється достовірність переданих даних, шляхом оцінювання інтенсивності бітових помилок (*BER*) у відновлених повідомленнях. Крім того, оцінюється величина внесених викривлень в контейнер-зображення. Для цього розраховується середньоквадратична помилка (*MSE*), між вихідним зображенням і тим, яке отримано після приховування в ньому інформаційного повідомлення. Розглянуті характеристики (*BER* і *MSE*) дозволяють порівняти різні варіанти формування розширюючих спектр сигналів. У роботі показано, що зміна правил формування сигналів може істотно впливати на BER.

Звичайно, в системах радіозв'язку з прямим розширенням спектру природний шум у КЗ не є корельований із розширюючими послідовностями. Однак, у разі приховування інформації в цифрові зображення це може бути не так. Найближчі (*сусідні*) пікселі реалістичних зображень досить сильно корельовані і цей зв'язок може істотно порушити базові припущення, що обумовлюють правильне відновлення прихованих інформаційних даних (*стеганоконтенту*). У цій статті досліджується декілька варіантів формування розширюючих сигналів і обґрунтовується вибір кращої альтернативи.

2 Технологія прямого розширення спектру в стеганографії

Передача даних у системах радіозв'язку з використанням технології прямого розширення спектру може бути спрощено представлена у вигляді співвідношення (1):

$$N = I + P\sum_{i=1}^{k} b_i\varphi_i, \tag{1}$$

де кожен інформаційний біт $b_i \in \{-1,1\}$ множиться на розширюючу псевдовипадкову послідовність φ_i із множини (ансамблю) слабокорелюючих дискретних сигналів:

$$\forall \varphi_i \in \varphi = \{\varphi_0, \varphi_1, ..., \varphi_{M-1}\}, \tag{1а}$$

$$\forall i \neq j : \rho(\varphi_i, \varphi_j) \approx 0, \tag{1б}$$

- P - коефіцієнт посилення потужності дискретних сигналів;
- k - число бітів інформаційного повідомлення, які передаються одночасно в каналі зв'язку (в системах кодового розподілу каналів ця величина може характеризувати абонентську ємність множинного доступу);
- I - природний шум в КЗ;
- $\rho(\varphi_i, \varphi_j)$ - коефіцієнт взаємної кореляції послідовностей φ_i та φ_j;
- N - отриманий на приймальній стороні сигнал (*адитивна суміш корисного сигналу та шуму*).

Відновлення інформації здійснюється за допомогою кореляційного прийому. Для цього обчислюється коефіцієнт кореляції (*скалярний добуток векторів*):

$$\rho\left(N, \varphi_j\right) = I\varphi_j + \varphi_j P\sum_{i=1}^{k} b_i\varphi_i \cdot \tag{1в}$$

У системах зв'язку природний шум і шумовий сигнал φ_i статистично незалежні (*некорельовані*), тобто $\rho\left(I, \varphi_j\right) = I\varphi_j \approx 0$. Різні шумові сигнали також некорельовані один з одним, отже $\forall j \neq i : \varphi_j\varphi_i \approx 0$. Тоді $\rho\left(N, \varphi_j\right) \approx Pb_j\varphi_j\varphi_j$ і значення b_j можна визначити за знаком $\rho\left(N, \varphi_j\right)$:

$$b_j = sign\left[\rho\left(N, \varphi_j\right)\right]. \tag{2}$$

Для приховування інформаційного повідомлення в контейнері-зображенні використовуються наступні припущення [5-11]. При цьому, цифрове зображення інтерпретується, як шум в КЗ, при цьому ми припускаємо, що $\rho\left(I, \varphi_j\right) = I\varphi_j \approx 0 \cdot$

Інформаційні біти модулюються розширюючими послідовностями: $\sum_{i=1}^{k} b_i\varphi_i$, після чого, так само як і в (1), посилений результат додається до контейнера-зображення.

Для відновлення інформаційних бітів, також, використовується правило (2).

Як і раніше, вважаємо $\forall j \neq i : \varphi_j \varphi_i \approx 0$. Однак припущення $\rho\left(I, \varphi_j\right) = I\varphi_j \approx 0$ може не виконуватися. Дійсно, окремі пікселі реалістичних зображень сильно корельовані. В цьому випадку результат $\rho\left(I, \varphi_j\right) = I\varphi_j$ залежить від статистичних властивостей розширюючих послідовностей, тобто від способу формування множини $\varphi = \{\varphi_0, \varphi_1, ..., \varphi_{M-1}\}$.

В цій статті розглядаються різні способи формування дискретних сигналів і досліджується ефективність їх використання для приховування повідомлення в контейнерах-зображеннях. Оцінюється інтенсивність бітових помилок (BER) при відновленні даних за правилом (2).

Показник BER – кількість бітових помилок N_{error} розділених на загальну кількість переданих N_{total} [23]:

$$BER = \frac{N_{error}}{N_{total}}. \tag{3}$$

Отже BER - критерій якості роботи, який виражається у відсотках [23]. В даній статті він оцінюється в абсолютних величинах, тобто безпосередньо по формулі (3).

В проведених дослідженнях BER оцінювалося без використання завадостійкого кодування. Цей випадок також розглядається в інших роботах, наприклад, в таблиці 2 з [8] наведено схожі результати.

Для оцінки викривлень контейнера-зображення використовують показник MSE [23-25]. Для монохромного $m \times n$ зображення I значення MSE визначають за формулою (4):

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I_{i,j} - N_{i,j}]^2, \tag{4}$$

де N - викривлене помилками наближення контейнера-зображення (*далі - контейнера*), як, наприклад, в (1).

В якості вихідних даних використовувалися різні 256×256 зображення (по аналогії з роботами [7-8, 10-11]). Наведені нижче результати становлять усереднені значення, які отримані за кількома різними зображеннями.

3 Результати досліджень

Розглянемо декілька варіантів формування розширюючих послідовностей в (1). Для кожного випадку будуть оцінені BER та MSE. Дані значення характеризують помилки у відновленому повідомленні та викривлення, що вносяться у контейнер.

Розглянемо декілька варіантів формування розширюючих послідовностей в (1). Для кожного випадку будуть оцінені BER та MSE. Дані значення характеризують помилки у відновленому повідомленні та викривлення, що вносяться у контейнер-зображення.

3.1 Нелінійні послідовності зі стандартним нормальним розподілом

Перший випадок, який буде розглянуто, описаний в роботах [7-8, 10-11], де пропонується формувати кожну розширюючу послідовність із використання відношень (5, 6):

$$\left(\varphi_i\right)_j = \begin{cases} \Phi^{-1}((u_i)_j), b_j = -1; \\ \Phi^{-1}((u_i')_j), b_i = 1, \end{cases} \tag{5}$$

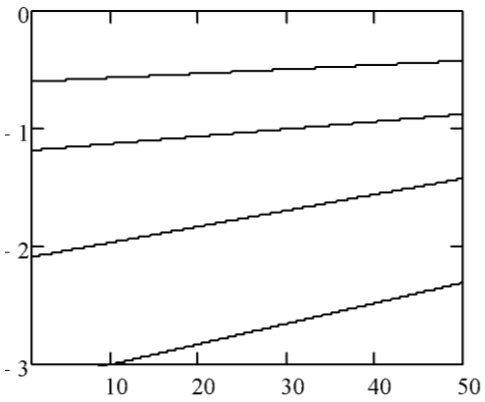
де

$$(u_i')_j = \begin{cases} (u_i)_j + 0.5, u_i < 0.5; \\ (u_i)_j - 0.5, u_i \geq 0.5, \end{cases} \tag{6}$$

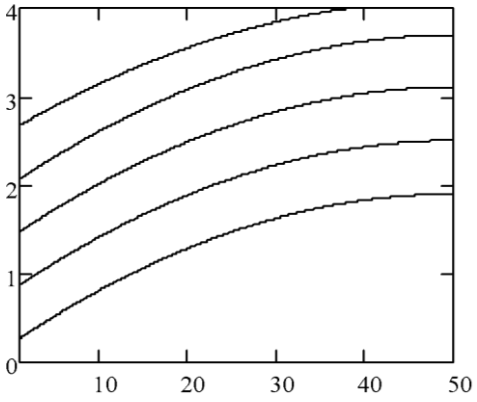
- $(u_i)_j$ - рівномірно розподілена на інтервалі (0,1) випадкова величина;

- Φ^{-1} представляє собою зворотну кумулятивну функцію розподілу для стандартної гаусової випадкової величини.

Отримані результати для різних P (для дискретних послідовностей із [7, 8, 10, 11]), наведені на рис. 1.



а) BER(k), P = 8, 16, 32, 64

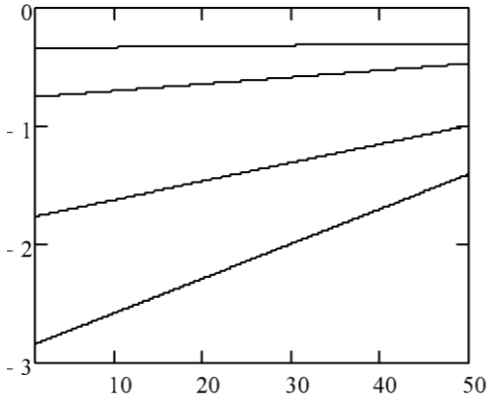


б) MSE(k), P = 1, 2, 4, 8, 16

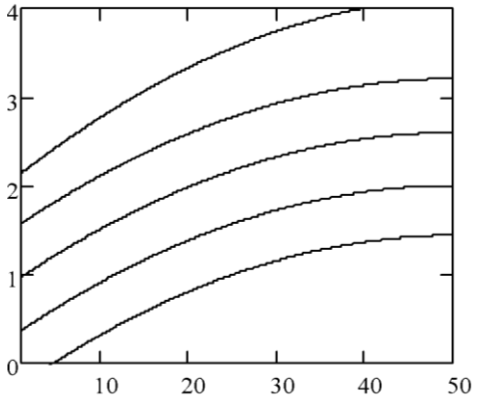
Рис. 1 – Емпіричні залежності BER(k) та MSE(k)

3.2 Дискретні послідовності з рівномірним на інтервалі (-1,1) розподілом

В цій роботі було досліджено ще кілька способів формування розширюючих послідовностей. Як альтернатива стандартному нормальному розподілу було реалізовано інший спосіб формування дискретних послідовностей, коли їхні елементи розподілені за рівномірним на інтервалі (-1,1) законом. Результати експериментальних досліджень, для дискретних послідовностей з рівномірно розподіленими на інтервалі (-1,1) значеннями, наведені на рис. 2.



а) BER(k), P = 8, 16, 32, 64



б) MSE(k), P = 1, 2, 4, 8, 16

Рис. 2 – Емпіричні залежності BER(k) та MSE(k)
(з рівномірно розподіленими на інтервалі (-1,1) значеннями)

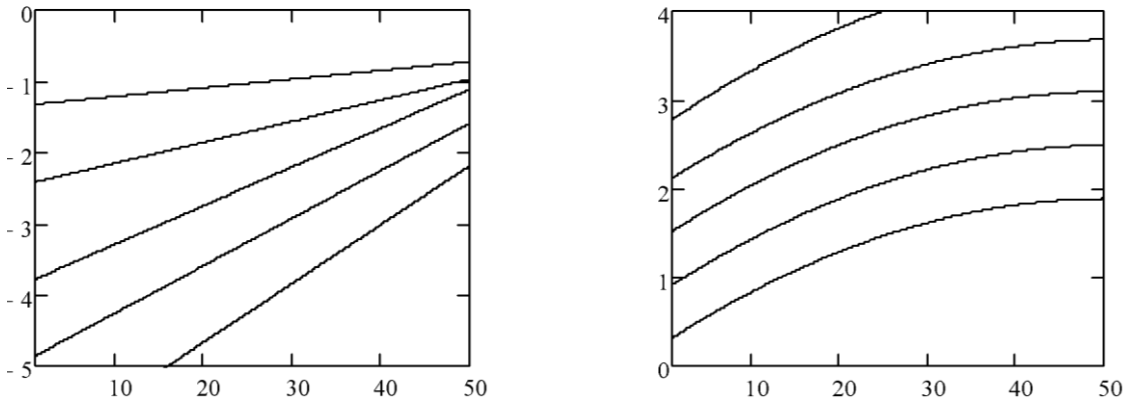
3.3 Ортогональні дискретні сигнали Уолша-Адамара

Ще один спосіб формування множини $\varphi = \{\varphi_0, \varphi_1, \ldots, \varphi_{M-1}\}$, який було досліджено, полягав у використанні матриць Адамара. Ці матриці формуються за рекурентним правилом (7):

$$H_{2^i} = \begin{bmatrix} H_{2^{i-1}} & H_{2^{i-1}} \\ H_{2^{i-1}} & -H_{2^{i-1}} \end{bmatrix}, \; H_1 = [1] \qquad (7)$$

При цьому, рядки (або стовбці) матриць H_{2^i} взаємно ортогональні, тобто їхній скалярний добуток дорівнює нулю. Множину дискретних сигналів $\varphi = \{\varphi_0, \varphi_1, \ldots, \varphi_{M-1}\}$, складену з таких рядків (або стовпців), називають послідовностями Уолша-Адамара [22].

Результати експериментальних досліджень BER і MSE для випадку для дискретних послідовностей (*сигналів*) Уолша-Адамара наведені на рис. 3.



а) BER(k), P = 1, 2, 4, 8, 16 *б) MSE(k), P = 1, 2, 4, 8, 16*

Рис. 3 – Емпіричні залежності BER(k) та MSE(k)
(для сигналів Уолша-Адамара)

4 Результати та висновки

За результатами проведених експериментальних досліджень можна побачити, що всі розглянуті способи генерації дискретних сигналів практично еквівалентні по викривленню зображення-контейнера. Це пояснюється близьким діапазоном можливих значень елементів послідовностей, та схожим способом приховування інформації. При цьому, невелику перевагу мають сигнали з нелінійним правилом модуляції, які були запропоновані в роботах [7-8 та 10-11]. Найгірше, за критерієм MSE, виглядають ортогональні дискретні сигнали Уолша-Адамара (*але цей програш невеликий і практично непомітний на логарифмічній шкалі*).

За критерієм мінімізації BER перші два способи формування дискретних сигналів практично однакові. Навіть при високому коефіцієнті посилення P ці методи формування розширюючих послідовностей не дають можливості отримати малі величини BER. Наприклад, навіть при $P = 64$ інтенсивність помилки приблизно дорівнює 10^{-3} і вище, що передбачає обов'язкове використання завадостійких кодів. Невелику перевагу серед перших двох способів мають нелінійні послідовності, що розглянуті в роботах [7-8 та 10-11]. Однак найбільш ефективним способом зі зниженням BER, є використання дискретних сигналів Уолша-Адамара.

Із діаграм, які наведено на рис. 3 слідує, що навіть при $P = 16$ вже досягаються низькі значення BER, що, приблизно, дорівнюють 10^{-5} та нижче. Це відкриває досить широкі можливості стосовно практичної побудови стеганографічних систем приховування інформації в контейнерах-зображеннях різного типу.

Багатообіцяючим напрямком подальших досліджень є розробка адаптивного алгоритму формування розширюючих псевдовипадкових послідовностей. Наприклад, якщо правило формування дискретних сигналів із множини $\varphi = \{\varphi_0, \varphi_1, \ldots, \varphi_{M-1}\}$ буде враховувати статистичні властивості зображення-контейнера, то інтенсивність помилок BER можна істотно знизити, та навидь домогтися безпомилкового відновлення інформації. В цьому сенсі перспективним напрямом слід вважати використання нових класів псевдовипадкових послідовностей, які запропоновані в роботах [26-30].

Посилання

[1] “Digital Watermarking and Steganography,” 2008. doi:10.1016/b978-0-12-372585-1.x5001-3.

[2] F. Y. Shin, “Digital Watermarking and Steganography,” Dec. 2017. doi:10.1201/9781315219783.

[3] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," in Computer, vol. 31, no. 2, pp. 26-34, Feb. 1998. doi: 10.1109/MC.1998.4655281 .

[4] I. V. S. Manoj, “Cryptography and Steganography,” International Journal of Computer Applications, vol. 1, no. 12, pp. 63–68, Feb. 2010. doi:10.5120/257-414.

[5] A. Z. Tirkel, C. F. Osborne and R. G. Van Schyndel, "Image watermarking-a spread spectrum application," Proceedings of ISSSTA'95 International Symposium on Spread Spectrum Techniques and Applications, Mainz, Germany, 1996, pp. 785-789 vol.2. doi: 10.1109/ISSSTA.1996.563231.

[6] J. R. Smith and B. O. Comiskey, “Modulation and information hiding in images,” Lecture Notes in Computer Science, pp. 207–226, 1996. doi:10.1007/3-540-61996-8_42.

[7] L. M. Marvel, C. G. Boncelet, R. Jr., and Charles T., “Methodology of Spread-Spectrum Image Steganography,” Jun. 1998. doi:10.21236/ada349102.

[8] L. M. Marvel, C. G. Boncelet and C. T. Retter, "Spread spectrum image steganography," in IEEE Transactions on Image Processing, vol. 8, no. 8, pp. 1075-1083, Aug. 1999. doi: 10.1109/83.777088.

[9] M. Kutter, “Performance Improvement of Spread Spectrum Based Image Watermarking Schemes through M-ary Modulation,” Lecture Notes in Computer Science, pp. 237–252, 2000. doi:10.1007/10719724_17.

[10] F. S. Brundick and L. M. Marvel, “Implementation of Spread Spectrum Image Steganography,” Mar. 2001. doi:10.21236/ada392155.

[11] Patent No.: US 6,557,103 B1, Int.Cl. G06F 11/30. Charles G. Boncelet, Jr., Lisa M. Marvel, Charles T. Retter. Spread Spectrum Image Steganography. Patent No.: US 6,557,103 B1, Int.Cl. G06F 11/30. – № 09/257,136; Filed Feb. 11, 1999; Date of Patent Apr. 29, 2003

[12] Fan Zhang, Bin Xu and Xinhong Zhang, "Digital image watermarking algorithm based on CDMA spread spectrum," 2006 12th International Multi-Media Modelling Conference, Beijing, 2006, pp. 4 pp.-. doi: 10.1109/MMMC.2006.1651359.

[13] T. T. Nguyen and D. Taubman, "Optimal linear detector for spread spectrum based multidimensional signal watermarking," 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, 2009, pp. 113-116. doi: 10.1109/ICIP.2009.5414121.

[14] E. Nezhadarya, Z. J. Wang and R. K. Ward, "Image quality monitoring using spread spectrum watermarking," 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, 2009, pp. 2233-2236. doi: 10.1109/ICIP.2009.5413955.

[15] S. Ghosh, P. Ray, S. P. Maity and H. Rahaman, "Spread Spectrum Image Watermarking with Digital Design," 2009 IEEE International Advance Computing Conference, Patiala, 2009, pp. 868-873. doi: 10.1109/IADCC.2009.4809129.

[16] H. O. Altun, A. Orsdemir, G. Sharma and M. F. Bocko, "Optimal Spread Spectrum Watermark Embedding via a Multistep Feasibility Formulation," in IEEE Transactions on Image Processing, vol. 18, no. 2, pp. 371-387, Feb. 2009. doi: 10.1109/TIP.2008.2008222.

[17] A. Samčović and M. Milovanović, "Robust digital image watermarking based on wavelet transform and spread spectrum techniques," 2015 23rd Telecommunications Forum Telfor (TELFOR), Belgrade, 2015, pp. 811-814. doi: 10.1109/TELFOR.2015.7377589.

[18] V. P. Ipatov, “Spread Spectrum and CDMA,” Mar. 2005. doi:10.1002/0470091800.

[19] “Introduction to CDMA Wireless Communications,” 2007. doi:10.1016/b978-0-7506-5252-0.x5001-7.

[20] “The Generalized CDMA,” CDMA: Access and Switching, pp. 1–28. doi:10.1002/0470841699.

[21] S. Hara and R. Prasad, "DS-CDMA, MC-CDMA and MT-CDMA for mobile multi-media communications," Proceedings of Vehicular Technology Conference - VTC, Atlanta, GA, USA, 1996, pp. 1106-1110 vol.2. doi: 10.1109/VETEC.1996.501483.

[22] S. S. Agaian, H. G. Sarukhanyan, K. O. Egiazarian, and J. Astola, “Hadamard Transforms,” Aug. 2011. doi:10.1117/3.890094.

[23] “Probability Theory of Bit Error Rate,” Optical Bit Error Rate, 2009. doi:10.1109/9780470545430.ch7.

[24] J. Korhonen and J. You, "Peak signal-to-noise ratio revisited: Is simple beautiful?," 2012 Fourth International Workshop on Quality of Multimedia Experience, Yarra Valley, VIC, 2012, pp. 37-38. doi: 10.1109/QoMEX.2012.6263880

[25] “Data Compression,” 2007. doi:10.1007/978-1-84628-603-2.

[26] Yu.V. Stasev, A.A. Kuznetsov, A.M. Nosik. “Formation of pseudorandom sequences with improved autocorrelation properties.” Cybernetics and Systems Analysis, vol. 43, Issue 1, pp. 1-11, January 2007. DOI: 10.1007/s10559-007-0021-2

[27] N.I.Naumenko, Yu.V.Stasev, A.A.Kuznetsov. “Methods of synthesis of signals with prescribed properties.” Cybernetics and Systems Analysis, vol. 43, Issue 3, pp. 321-326, May 2007. DOI: 10.1007/s10559-007-0052-8

[28] O.Karpenko, A.Kuznetsov, V.Sai, Yu.Stasev. “Discrete Signals with Multi-Level Correlation Function.” Telecommunications and Radio Engineering, vol. 71, 2012 Issue 1. pp 91-98. DOI: 10.1615/TelecomRadEng.v71.i1.100

[29] A. Kuznetsov, S. Kavun, V. Panchenko, D. Prokopovych-Tkachenko, F. Kurinniy and V. Shoiko, "Periodic Properties of Cryptographically Strong Pseudorandom Sequences," 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2018, pp. 129-134. doi: 10.1109/INFOCOMMST.2018.8632021

[30] A. Kuznetsov, O. Smirnov, D. Kovalchuk, A. Averchev, M. Pastukhov and K. Kuznetsova, "Formation of Pseudorandom Sequences with Special Correlation Properties," 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), Lviv, Ukraine, 2019, pp. 395-399. doi: 10.1109/AIACT.2019.8847861

Reviewer: Mikołaj Karpiński, Dr. of Sciences (Eng.), Full Prof., University of Bielsko-Biala, Bielsko-Biala, Poland.
E-mail: mkarpinski@ath.bielsko.pl

Received: January 2020.

Authors:
Oleksii Smirnov, Central Ukrainian National Technical University, Cybersecurity & Software Academic Department, Kropivnitskiy, Ukraine. E-mail: dr.smirnovoa@gmail.com
Anna Arischenko, computer science student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: annaarischenko@gmail.com
Eugene Demenko, computer science student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: demenjay@gmail.com
Alexander Onikiychuk, computer science student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: onik4524a@gmail.com
Alexandr Kuznetsov, V. N. Karazin Kharkiv National University, Department of information systems and technologies security, Kharkiv, Ukraine. E-mail: kuznetsov@karazin.ua

Pseudorandom Sequences for Spread Spectrum Image Steganography.

Abstract. We consider pseudorandom sequences (signals), which are used for information-hiding in cover images. Spread spectrum image steganography is used for the hiding, the essence of which is modulating information data with long pseudorandom (noise) sequences. Messages take the form of noise, and it is extremely difficult to detect such transmission. We investigate different ways of discrete signals generation and estimate the error rate in message restoration. It appears, the way of discrete signals generation influences on the error rate and we prove the choice of the most suitable signals. Moreover, we estimate distortions of the cover image as a result of data-hiding. The article mainly contains the results of experimental researches, which can be useful in justifying various ways of building direct spread spectrum steganographic systems.

Keywords: Data-hiding; Steganography; Spread spectrum image steganography; Pseudorandom sequences; Spreading sequences.

Рецензент: Николай Карпинский, д.т.н., проф., Университет Бельсько-Бяла, ул. Виллова 2, 43-309 Бельсько-Бяла, Польша.
E-mail: mkarpinski@ath.bielsko.pl

Поступила: Январь 2020.

Авторы:
Алексей Смирнов, каф. кибербезопасности и программного обеспечения, Центральный украинский национальный технический университет, Кропивницкий, Украина.
E-mail: dr.smirnovoa@gmail.com
Анна Арищенко, студентка факультета компьютерных наук, ХНУ имени В. Н. Каразина, Харьков, Украина.
E-mail: annaarischenko@gmail.com
Евгений Деменко, студент факультета компьютерных наук, ХНУ имени В. Н. Каразина, Харьков, Украина.
E-mail: demenjay@gmail.com
Александр Оникийчук, студент факультета компьютерных наук, ХНУ имени В. Н. Каразина, Харьков, Украина.
E-mail: onik4524a@gmail.com
Александр Кузнецов, д.т.н., проф., каф. безопасности информационных систем и технологий, Харьковский национальный университет имени В. Н. Каразина, Харьков, Украина.
E-mail: kuznetsov@karazin.ua

Псевдослучайные дискретные последовательности для сокрытия данных в контейнерах-изображениях с использованием технологии прямого расширения спектра.

Аннотация. В статье рассматриваются псевдослучайные дискретные последовательности (сигналы), которые используются для стеганографического сокрытия информационных сообщений в контейнерах-изображениях. Для сокрытия применяется технология прямого расширения спектра, суть которой состоит в моделировании информационных данных длинными псевдослучайными (шумовыми) последовательностями. Сообщения приобретают вид шума, и обнаружить такую передачу крайне сложно. В работе исследуются различные способы формирования дискретных сигналов, и оценивается интенсивность ошибок при восстановлении сообщений. Оказывается, способ формирования псевдослучайных последовательностей влияет на интенсивность ошибок, а также в работе обосновывается выбор наиболее подходящих сигналов. Кроме того, оцениваются искажения контейнера-изображения в результате сокрытия данных. В статье содержатся преимущественно результаты экспериментальных исследований, которые могут быть полезны при обосновании различных вариантов построения стеганографических систем с прямым расширением спектра.

Ключевые слова: сокрытие информации; стеганография; технология прямого расширения спектра; псевдослучайная последовательность; расширяющие сигналы.

УДК 004.056.5 DOI: 10.26565/2519-2310-2019-4-02

РОЗРАХУНОК ЙМОВІРНОСТІ УСПІХУ АТАКИ РОЗГАЛУЖЕННЯ БЛОКЧЕЙН РЕЄСТРУ

Владислав Сафоненко, Микита Гончаров, Сергій Даценко, Єлизавета Лазарєва, Микола Полуяненко

Харківський національний університет імені В.Н. Каразіна, Харків, Україна
vladyslavsafonenko@gmail.com, wdpgames@yandex.ru, sergdacenko@gmail.com, lazareva15elizaveta@gmail.com, nlfsr01@gmail.com

Рецензент: Володимир Хома, д.т.н., проф., Опольський політехнічний Університет, Ополе, Польща
xoma@wp.pl

Поступила: Декабрь 2019.

Анотація: *У роботі систематизовано відомості за тематикою питань атаки розгалуження блокчейн реєстру. Запропоновано огляд та узагальнення інформації, яка представлена в найбільш авторитетних роботах за даним напрямом. Здійснено аналіз відповідних робіт стосовно оцінки ймовірностей подвійної витрати в протоколі консенсусу "Доказу виконаної роботи". Розглянуто проблематику розорення гравця та проведено аналогію з атакою подвійної витрати на блокчейн. Розглянуто експеримент Пуассона для загального випадку. Проаналізовано моделі на підставі яких, С. Накамото та М. Розенфельдом були зроблені спроби отримати кількісну оцінку ймовірності успішної атаки подвійної витрати на деякі алгоритми консенсусу, що мають ймовірнісну завершеність. Наведено спрощення і допущення, що мають місце у відповідних моделях, за допомогою яких отримано кінцевий вираз.*

Ключевые слова: *комп'ютерні мережі; децентралізовані системи; блокчейн технології; атака на блокчейн мережі; атака розгалуження, атака подвійної витрати, експеримент Пуассона.*

1 Вступ

Незважаючи на постійно зростаючу популярність блокчейн систем, кількість організацій, що впровадили його, все ще залишається відносно невеликим. Багато хто стурбований погрозами блокчейн технологій з точки зору безпеки, інші вважають, що технологія має повільний спосіб підтвердження транзакцій (*в тому числі здійснення платежів*). Ті, хто приймає його, повинні спробувати прийняти всі запобіжні заходи, перш ніж приймати транзакцію, щоб запобігти атакам з подвійною витратою.

Один з важливих запобіжних заходів полягає в тому, щоб вирішити, коли приймати транзакцію, перш ніж здійснювати операцію. Суб'єкти, що застосовують блокчейн технології, вважають за краще отримувати певну ступінь впевненості в якості гарантії того, щодо неможливості скасування прийнятої транзакції. Ті, хто може дозволити собі почекати тривалий період часу, перш ніж приймати транзакцію (*наприклад, онлайн-платформи*), вимагають, як мінімум шість підтверджень, перш ніж приймати транзакцію і вважати її необоротною. Однак інші, які не можуть дозволити собі цей час очікування (*наприклад, торгові автомати та системи, що працюють в онлайн режимі*), приймають транзакції з ризиком втрати платежу в результаті вдалої атаки подвійної витрати. Ймовірності успішної проведення атаки, що реалізує даний спосіб, присвячена досить велика кількість робіт. Далі ми проаналізуємо найбільш авторитетні та популярні роботи даного напрямку, які аналізують ймовірність успіху зловмисника на основі частки обчислювальної потужності, яку він контролює.

2 Ймовірність, що запропонована Сатоши Накамото

2.1 Припущення, викладені в роботі

Перша спроба зробити оцінку даної ймовірності наводиться в розділ 11 відповідної роботи Сатоши Накамото [1]. Стисло наведемо цей розділ з деякими нашими коментарями.

«Розглянемо сценарій, в якому зловмисник намагається генерувати більш довгий ланцюг блоків, ніж чесні учасники. Навіть, якщо він досягне успіху, це не призведе до того, що можна буде створювати «гроші з повітря», привласнювати собі чужі монети, або вносити інші довільні зміни. Вузли ніколи не приймуть некоректну транзакцію, навіть якщо блок її

містить. Атакуючий може лише намагатися змінити одну зі своїх транзакцій, щоб повернути собі гроші». Відносно даного твердження слід зауважити, що вузли не приймуть некоректну транзакцію тільки, якщо мають коректну програмну реалізацію всіх можливих перевірок (*приклад існуючої некоректної реалізації в мережі Bitcoin описаний в* [2, 3]), а також, якщо вузол не є вузлом зловмисника або не входить з ним у змову.

«Гонку між чесними учасниками і нападаючим можна уявити як біноміальне випадкове блукання. Успішна подія, коли «чесний» ланцюг подовжується на один блок, призводить до збільшення відриву на одиницю, збільшуючи свою перевагу на +1 , а неуспішне, коли черговий блок формує зловмисник, – до його скорочення на один блок, зменшуючи розрив на -1. Ймовірність атакуючого наздогнати різницю в кілька блоків така ж, як і в задачі про «розорення гравця». Уявімо, що гравець має необмежений кредит, починає з деяким дефіцитом і у нього є нескінченно багато спроб, щоб відіграться».

Таким чином, у цьому абзаці, моделювання здійснюється за допомогою наступних припущень:

Припущення С. Накамото № 1 – гонку між чесними учасниками і нападаючим можна уявити, як біноміальне випадкове блукання.

Припущення С. Накамото № 2 – ймовірність перемоги зловмисника еквівалентна задачі про «розорення гравця».

Припущення С. Накамото № 3 – зловмисник має можливість (бажання) нескінченно довго проводити атаку на мережу формуючи альтернативний ланцюжок. Це припущення детально розглядається в роботі [4].

«Ймовірність того, що він досягне успіху, як і ймовірність зловмисника наздогнати чесних учасників, обчислюється таким чином [5]:

$p =$ ймовірність появи блоку у чесному ланцюжку;

$q =$ ймовірність того, що блок створить атакуючий;

$q_z =$ ймовірність того, що атакуючий надолужить різницю в z блоках:

$$q_z = \begin{cases} 1 & якщо, p \le q \\ (q/p)^z & якщо, p > q \end{cases}. \quad (1)$$

В разі $p > q$ ймовірність зменшується експоненційно з ростом числа блоків, на яке відстає зловмисник. Оскільки всі ставки проти нього, без вдалого ривка на початку його шанси на успіх стають мізерно малі».

Припущення С. Накамото № 4 – ймовірності формування блоку чесною мережею або зловмисником вважаються константами, що не змінюються у часі. Однак, як зазначено у роботах [4, 6] чесна мережа або зловмисник можуть додати обчислювальних потужностей, або навпаки, з плином часу вони можуть зменшитися. Пінзон та Роча пропонують для рівняння, що керує цими моделями, використовувати розподіл ймовірностей Ерланга (*на відміну від Накамото, що використовує розподіл ймовірностей Пуассона та Розенфельда, котрий використовує від'ємний біноміальний розподіл ймовірностей*).

«Розглянемо тепер, як довго одержувачу платежу варто чекати, перш ніж він буде повністю впевнений, що колишній власник не зможе скасувати транзакцію. Ми припускаємо, що зловмисник-відправник дозволяє адресату деякий час вірити, що платіж був проведений, після чого повертає гроші собі. Одержувач дізнається про це, але шахрай сподівається, що буде вже занадто пізно.

Адресат створює нову пару ключів і повідомляє свій публічний ключ відправнику прямо перед підписанням транзакції. Це не дозволить відправнику заздалегідь почати працювати над ланцюжком і провести транзакцію в той момент, коли він буде досить вдалим, щоб зробити ривок вперед. Після відправки платежу шахрай починає потай працювати над паралельною версією ланцюжка, що містить альтернативну транзакцію».

Припущення С. Накамото № 5 – одержувач створює новий гаманець (*пару ключів*) безпосередньо перед підписанням транзакції. Як показує практика, багато інтернет магазинів або

користувачів мають фіксовані біткойн-адреси, що є добре відомими і загальнодоступними протягом досить тривалого (*у порівнянні з часом, необхідним на проведення атаки подвійний витрати*) часу.

«Одержувач чекає, поки транзакція не буде додана в блок і поки той не буде продовжений ще блоками. Йому невідомий прогрес зловмисника, але якщо середня швидкість генерації чесних блоків – відома величина, то число блоків нападника підпорядковується розподілу Пуассона з математичним очікуванням: $\lambda = z\frac{q}{p}$».

Припущення С. Накамото № 6 – функція прогресу зловмисника відповідає розподілу Пуассона. У роботі [7] показано помилковість даного припущення С. Накамото та і наводиться більш влучний вираз, відповідно до негативного біноміального розподілу.

«Щоб отримати можливість того, що атакуючий обжене чесних учасників мережі у кількості створених блоків, ми множимо значення випадкової величини (*число створених ним блоків*) на ймовірність того, що він зможе надолужити різницю, що залишилася (2):

$$\sum_{k=0}^{\infty}\frac{\lambda^k e^{-\lambda}}{k!}\cdot\begin{cases}(q/p)^{(z-k)} & \text{если } k \le z\\ 1 & \text{если } k > z\end{cases}. \qquad (2)$$

Перегрупувавши складові і позбавляючись від нескінченної низки, отримуємо (3):

$$1-\sum_{k=0}^{z}\frac{\lambda^k e^{-\lambda}}{k!}\left(1-(q/p)^{(z-k)}\right). \qquad (3)$$

Склавши відповідну програму (*для розрахунку ймовірності того, що атакуючий зможе надолужити різницю (p)*), та проаналізувавши отримані результати, можливо побачити, що ця ймовірність експоненційно падає з ростом z (див. рис. 1-2).

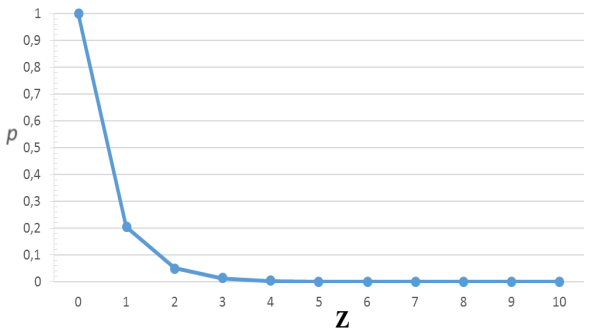


Рис. 1 – Розподіл ймовірності *p* при *q=0,1**

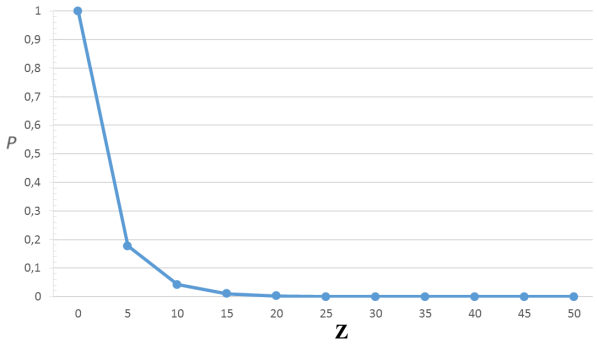


Рис. 2 – Розподіл ймовірності *p* при *q=0,3**

** q - ймовірність того, що блок створить атакуючий.*

2.2 Аналіз результатів щодо інших обмежень

Для того, щоб продемонструвати, які ще спрощення та допущення були зроблені в роботі С. Накамото, необхідно більш детально розглянути виведення математичних виразів, на які саме спирався С. Накамото при створенні моделі атаки. Більшість з наведеного в цьому підрозділі взято з [8, 9]. Відповідно до припущень С. Накамото № 1, гонку між чесними учасниками та нападаючим можна уявити, як біноміальне випадкове блукання.

Випадкове блукання – це математичний процес, який відбувається уздовж ряду станів, що з'єднані лінією (Рис. 3). Кожний стан нумерується, а процес починається зі стану «0». Таким чином, підкидаючи монету, ми просуваємося вперед по «орлам» і назад по «решкам» уздовж ряду станів.

Біткойн налаштовується так, що блоки виявляються приблизно кожні десять хвилин. При спробі реалізації атаки подвійної витрати, атакуючий буде генерувати блок в середньому кожні $10/q$ хвилин, а чесні майнери – генерувати блок в середньому кожні $10/p$ хвилин. Але, це тільки в середньому. Через випадковості, що притаманні майнінгу, зловмисник в будь-який момент може згенерувати на кілька блоків більше або менше, ніж чесні майнери. «Наше» випадкове блукання буде відслідковувати відмінність між їхніми підрахунками.

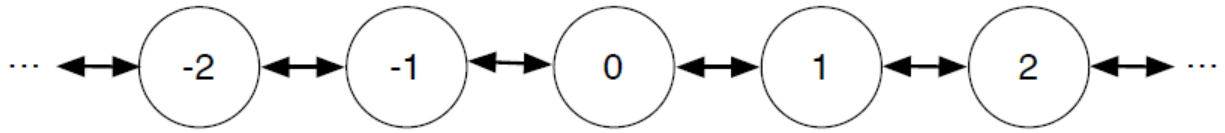


Рис. 3 – Схема процесу підрахунку

Щоб зрозуміти сутність походження наведеного вище рівняння (1), для якого Накамото цитує підручник Феллера 1968 року [5], ми повинні декілька заглибитися в проблему «завдання про розорення гравця» та її «невелику зміну». Для цього трохи змінимо позначення, які використовує Накамото, щоб зробити вираження більш простішим і послідовнішим. Те, що Накамото (і Феллер) позначили як «q_z» у формулі (1) в оригінальній статті, ми позначаємо як «Q_z».

2.3 Задача «розорення гравця»

Ця «знаменита» задача вперше була досліджена Блезом Паскалем (Blaise Pascal) і П'єром де Ферма (Pierre de Fermat) в 1656 році [10]. Вона моделює гравця, який входить в казино, щоб зіграти в просту азартну гру. Задача стартує з початкового стану з i монет та робить серію ставок. Кожна ставка приводить або до виграшу 1 монети з ймовірністю q, або програшу 1 монети з ймовірністю $p = q - 1$. Виграш або програш при кожній ставці не залежить від всіх інших ставок. Мета гравця полягає в тому, щоб виграти N монет, перш ніж розоритися (*тобто зменшити свій капітал до 0 монет*). Якщо гравець розорився, він більше не зможе грати, тому що у нього нема грошей, щоб виплатити 1 монету в разі наступного програшу. Таким чином, досягнення N або 0 завершує гру.

Припущення С. Накамото № 7 – $p + q = 1$. Тут і далі використовуються умови взаємозв'язку ймовірності формування блоку зловмисником та чесною мережею, в загальному ж випадку ймовірності q та p є незалежними значеннями. Наведені вирази не дають відповіді яким саме буде результат при незалежних величинах цих ймовірностей, що було зазначено у [11].

Дійсно, у визначенні завдання про розорення гравця використовується ймовірнісний простір з двома елементарними подіями: - «виграв перший гравець» та «виграв другий гравець». При моделюванні атаки подвійної витрати С. Накамото (*і, як буде показано далі, М. Розенфельд*) інтерпретують елементарні результати цього завдання як «блок, сформований чесною мережею» (*з ймовірністю такого результату* p) та «блок, сформований атакуючим» (*з ймовірністю* q), при чому $p = 1 - q$.

Однак в реальних блокчейн-системах ймовірність формування блоку (*знаходження прообразу функції гешування*) визначається виключно гешрейтом (*обчислювальними можливостями*) кожного учасника, тобто умова $p = 1 - q$ не повинна завжди виконуватися.

2.4 Невелика варіація розорення гравця

Щоб повернутися до С. Накамото, нам потрібно змінити гру. Накамото стверджує про ймовірність того, що зловмисник «коли-небудь наздожене», що є досить рішучою заявою. При цьому Накамото не аналізує, чи вдасться вирішити економіку: - можливо, зловмисник витрачає більше на майнінг, ніж вийде повернути після успішного проведення їм атаки подвійної витрати, або можливо, винагорода у вигляді монет за анонсування величезної кількості нових блоків більш прибуткова, ніж подвійна витрачена транзакція.

Метою Накамото є *виключно аналіз найгіршого сценарію*, коли зловмисник не шкодує витрат на використання своїх існуючих потужностей майнингу, щоб виграти в грі «розорення гравця». В контексті цього Накамото, можливо, намагався визначити найгірший випадок для кожного значення q, але в дійсності, найгіршим випадком є просто будь-яке значення $q > 0{,}5$.

Щоб досягти анонсованої мети Накамото, ми спочатку дозволимо атакуючому втратити до «y» монет, перш ніж виграти (*і потім ми подивимося, що станеться, коли y перейде в нескінченність*). Таким чином, ця невелика зміна перетворюється у початкове розорення гравця в такий спосіб: – гравець починає з $i = y$ монет, а гра закінчується або при 0 монетах, що є поразкою, або при $N = y + z$ монетах, які представляють собою виграш

$$q_y = \begin{cases} \dfrac{1-(p/q)^y}{1-(p/q)^{y+z}}, & \text{якщо } p \neq q; \\ \dfrac{y}{(y+z)}, & \text{якщо } p = q = 0{,}5. \end{cases} \tag{4}$$

Розглянемо випадок, коли гравець хоче втратити нескінченну суму грошей та, отже, має при цьому всі необмежені ресурси. Іншими словами, коли «y» йде у нескінченність. У разі, коли, $p < q$, то $(p/q)^y \to 0$, так як $y \to \infty$:

$$\lim_{y\to\infty} \frac{1-(p/q)^y}{1-(p/q)^{y+z}} = 1, \text{ коли } p < q. \tag{5}$$

У разі, коли $p > q$, для розрахунку границь беремо співвідношення $(p/q)^y$, як коефіцієнт з чисельника і знаменника:

$$\frac{1-(p/q)^y}{1-(p/q)^{y+z}} = \frac{(p/q)^y\left((p/q)^{-y}-1\right)}{(p/q)^y\left((p/q)^{-y}-(p/q)^z\right)} = \frac{(p/q)^{-y}-1}{(p/q)^{-y}-(p/q)^z}. \tag{6}$$

Відповідно, коли $p > q$, то $\left(\frac{p}{q}\right)^{-y} = \left(\frac{q}{p}\right)^y \to 0$, так як $y \to \infty$:

$$\lim_{y\to\infty} \frac{(p/q)^{-y}-1}{(p/q)^{-y}-(p/q)^z} = \frac{-1}{-(p/q)^z} = \left(\frac{q}{p}\right)^z, \text{ коли } p > q. \tag{7}$$

Оскільки в рівнянні (7) передбачається, що зловмисник має необмежені ресурси, то ми не можемо використовувати наш існуючий запис («q_∞» насправді не має сенсу), тому ми змінимо запис, та дозволимо «Q_z» позначати ймовірність надолуження з урахуванням z при необмежених ресурсах:

$$Q_z = \begin{cases} 1, & \text{якщо } p \le q; \\ \left(\dfrac{q}{p}\right)^z, & \text{якщо } p > q. \end{cases} \tag{8}$$

Слід зазначити, що вираз (8) справедливий тільки, якщо зловмисник має необмежені ресурси (*тобто, $N = y + z = [y \to \infty] = \infty$*), що вже обмовлялося вище (*див. Припущення Сатоши Накамото № 3*).

Рівняння для цього розділу було адаптовано з відповідних заміток Л. Рей-Беллі [12].

2.5 Аналогія з атакою подвійної витрати на блокчейн

Аналогію з нашим сценарієм блокчейну слід провести наступним чином. Нехай p – потужність майнингу та ймовірність того, що чесні майнери знайдуть наступний блок, а q – це

потужність майнінгу атакуючого. Ми визначаємо $q + p = 1$, так як припускаємо, що тільки атакуючий або чесний майнер може сформувати блок в кожному раунді. Якщо у атакуючого для здійснення майнінгу є необмежені апаратні ресурси та він зупиняється, коли досягає z, то ми можемо використовувати рівняння (8).

Припущення С. Накамото № 8 – тут варто відзначити, що «Q_z» це ймовірність того, що зловмисник просто наздожене чесну мережу. Замість цього Накамото мав би вирахувати «Q_{z+1}», тобто ймовірність того, що атакуючий випередить чесних майнерів.

Таким чином, у виразі (4) необхідно визначити виграш, як $N = y + z + 1$ [9]. Це припущення змінює гру так, щоб вона враховувала ймовірність атакуючого перевершити блоки, zrs сформовані чесними майнерами (9):

$$q_y = \begin{cases} \dfrac{1-(p/q)^y}{1-(p/q)^{y+z+1}}, & \text{якщо } p \neq q; \\ \dfrac{y}{(y+z+1)}, & \text{якщо } p = q = 0{,}5. \end{cases} \tag{9}$$

При цьому, якщо $q > p$, то це не призведе до змін, бо:

$$\lim_{y\to\infty} \frac{1-(p/q)^y}{1-(p/q)^{y+z+1}} = 1, \tag{10}$$

але для $p > q$, розділивши чисельник та знаменник на $(p/q)^y$, а потім обчислюючи границю отримуємо, що:

$$\frac{1-(p/q)^y}{1-(p/q)^{y+z+1}} = \frac{(p/q)^y\left((p/q)^{-y}-1\right)}{(p/q)^y\left((p/q)^{-y}-(p/q)^{z+1}\right)} = \frac{(p/q)^{-y}-1}{(p/q)^{-y}-(p/q)^{z+1}} \tag{11}$$

$$\lim_{y\to\infty} \frac{(p/q)^{-y}-1}{(p/q)^{-y}-(p/q)^{z+1}} = \frac{-1}{-(p/q)^{z+1}} = \left(\frac{q}{p}\right)^{z+1}, \quad (\text{коли } p > q).$$

Все це призведе до зміни виразу (8) до наступного виду:

$$Q_z = \begin{cases} 1, & \text{якщо } p \leq q; \\ \left(\dfrac{q}{p}\right)^{z+1}, & \text{якщо } p > q. \end{cases} \tag{12}$$

2.6 Експеримент Пуассона

Сатоши Накамото продовжує наступний аналіз.

«Одержувач чекає, поки транзакція не буде додана в блок та поки той не буде продовжений ще «z» блоками. Йому невідомий прогрес зловмисника, але якщо середня швидкість генерації чесних блоків – відома величина, то число блоків нападника підпорядковується розподілу Пуассона з математичним очікуванням:

$$\lambda = z\frac{q}{p}». \tag{13}$$

Щоб знайти це очікуване значення, Накамото використовує математичну модель, яка має назву «експеримент Пуассона» (*Припущення С. Накамото № 6*). В експерименті Пуассона ми моделюємо реальну ситуацію, що пов'язана з ймовірністю, підраховуючи кількість успіхів в серії інтервалів, виміряних в часі.

Щоб використовувати таку модель, ми повинні припустити наступне [13]:

1. *Кількість успіхів протягом кожного часового інтервалу не залежить від будь-якого іншого інтервалу.*
2. *Ймовірність того, що один успіх відбудеться протягом дуже короткого інтервалу часу, пропорційна тривалості інтервалу часу.*
3. *Ймовірність більш ніж одного успіху за такий короткий проміжок часу незначна.*
4. *Ймовірність успіху не змінюється під час експерименту, хоча в дійсності майнер може збільшувати або зменшувати свої ресурси (Припущення С. Накамото № 4).*

Щоб використовувати добре відомі результати для пуассонівських експериментів, наша перша задача – визначити значення «λ», яке є середнім числом успіхів, які ми очікуємо протягом кожного інтервалу. Це показник: – «*успіхи / інтервал*».

Для нас успіх – це кількість блоків, які, ми припускаємо, знайде зловмисник. А інтервал – це час, витрачений торговцем на очікування формування «*z*» блоків чесними майнерами.

Мережа біткоін налаштована таким чином, що кожні $T = 10$ хвилин виявляється 1 блок з 100% поточної потужності майнінгу. Для чесних вузлів кожні «*T*» хвилин виявляється «*p*» блоків. Щоб отримати «*z*» блоків, їм знадобиться наступний інтервал:

$$z \text{ блоків} \cdot \frac{T \text{ хвилин}}{p \text{ хвилин}} = \frac{zT}{p} \text{ хвилин}. \tag{14}$$

Для атакуючого q блоків виявляються кожні T хвилин, тому протягом цього інтервалу зловмисник буде формувати блоки зі швидкістю (15):

$$\lambda = \left(\frac{zT}{p} \text{хвилин/интервал} \right) \cdot \frac{q \text{ блоків}}{T \text{ хвилин}} = \frac{zq}{p} \text{ блоків/интервал}, \tag{15}$$

де $\lambda = \frac{zq}{p}$, просто середнє. Це означає, що випадкова величина була замінена на її математичне сподівання.

Припущення С. Накамото № 9 – (*як було зазначено в роботі* [14]) випадкову величину кількості сформованих було замінено на математичне очікування цієї величини.

У випробуванні пуассонівського експерименту ми будемо отримувати дані з розподілу Пуассона (*тобто ми будемо кидати кубик з розподілом Пуассона*), щоб побачити, скільки вдалих випробувань насправді сталося. Припустимо, сталося «*X*» вдач у конкретному випробуванні. Ймовірність того, що $X = k$ успіхів відбулося протягом нашого інтервалу, де $k \geq 0$, успіхів, які відбулися протягом нашого інтервалу, де:

$$P(X = k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}. \tag{16}$$

Рівняння (16) називається «функцією щільності ймовірності Пуассона» [13].

2.7 Загальний випадок

Ми хочемо знати відповідь на більш загальне питання: - враховуючи, що продавець буде чекати *«z»* блоків, перш ніж фізично передати товари, замовлені зловмисником, наскільки ймовірним є те, що зловмисник з потужністю майнінгу q може виробити більше блоків, ніж чесні майнери до цього моменту або після?

Відповідь Накамото полягає в наступному. Нехай *«X»* буде випадковою величиною, що представляє кількість блоків, які зловмисник знайде за час, коли чесні майнери сформували *«z»* блоків. Ми вже визначили $P(X; \lambda)$, як ймовірність того, що атакуючий сформує *«X»* блоків. Також ми знаємо, що ймовірність надолуження від різниці, що залишилася $z - k$ дорів-

нює q_{z-k}. Тому, щоб знайти загальну ймовірність надолуження, ми підсумовуємо всі можливості X:

$$P\left(X=0;\lambda\right)Q_z + P\left(X=1;\lambda\right)Q_{z-1} + \ldots = \sum_{k=0}^{\infty} P\left(X=k;\lambda\right)Q_{z-k} =$$

$$= \sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} Q_{z-k} \cdot \tag{17}$$

Фактично, коли $k > z$, ймовірність того, що зловмисник наздожене, дорівнює 1.

І так, як стверджує Сатоши Накамото: – «Щоб отримати можливість того, що атакуючий обжене чесних учасників, ми множимо значення випадкової величини (*число створених ним блоків*) на ймовірність того, що він зможе надолужити різницю, що залишилася(19):

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} (q/p)^{(z-k)} & \text{якщо } k \le z \\ 1 & \text{якщо } k > z \end{cases}\text{».} \tag{18}$$

Нарешті, оскільки ймовірність того, що щось трапиться, дорівнює 1 мінус ймовірність того, що це не так, Накамото перебудовує наш (*майже*) кінцевий результат. Тут віднімаємо з 1 ймовірність того, що атакуючий видобуває k блоків і не наздожене (19)

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} (q/p)^{(z-k)}, \text{якщо } k \le z \\ 1 \quad , \text{якщо } k > z \end{cases} = 1 - \sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} 1-(q/p)^{(z-k)}, \text{якщо } k \le z \\ 1-1 \quad , \text{якщо } k > z \end{cases} =$$

$$= 1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \left(1 - \left(\frac{q}{p}\right)^{(z-k)}\right) - \sum_{k=z+1}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot (0) = 1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \left(1 - \left(\frac{q}{p}\right)^{(z-k)}\right). \tag{19}$$

Або, як Накамото говорить лаконічніше: «Перестановка, щоб уникнути підсумовування в нескінченному ряду розподілу

$$1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - (q/p)^{(z-k)}\right)\text{».} \tag{20}$$

Знову ж таки, з огляду на *«Припущення С. Накамото № 8»* ми зацікавлені в тому, щоб зловмисник випередив чесних майнерів (21)

$$1 - \sum_{k=0}^{z+1} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - (q/p)^{(z+1-k)}\right). \tag{21}$$

2.8 Ймовірність, що запропонована Мені Розенфельдом

Наведемо, з деякими нашими коментарями, витяг з розділу 4 роботи Розенфельда [15]. В авторському тексті ми зминемо оригінальні позначення, що введені Мені Розенфельдом на позначення, які ми вже використовували раніше в даній роботі.

«Стандартною практикою для продавця є очікування z підтверджень платіжної транзакції, а потім надання продукту. Поки мережа формує ці підтверджуючі блоки, зловмисник будує свою власну гілку, яка суперечить загальнодоступному ланцюгу. Яка ймовірність, що він вдало проведе атаку подвійної витрати?

До отримання z підтверджень зловмисник не може опублікувати своє альтернативне розгалуження, навіть якщо воно довше, оскільки він відрадить продавця від виконання замовлення. Він повинен дочекатися z підтверджень, а вже потім, або опублікувати свою гілку,

якщо у нього є перевага, або продовжити роботу над нею, сподіваючись, що він отримає необхідну перевагу.

Шанси на успіх у вирішальній мірі залежать від відставання зловмисника в момент досягнення z підтверджень чесною мережею. У своїй статті Сатоши Накамото робиться спрощене припущення (*згадане нами, як Припущення С. Накамото № 9*), що чесна мережа за середній час знаходить z блоків, $\frac{zT}{p}$ та, відповідно, k – число блоків знайдених атакуючим за цей час, слідує розподілу Пуассона (*Припущення С. Накамото № 6*) із середнім $z\frac{q}{p}$. Ми не будемо використовувати це припущення, а будемо більш точно моделювати k, як від'ємну біноміальну змінну; це кількість успіхів (*блоків, які виявив атакуючий*) до z невдач (*блоків, виявлених чесною мережею*) з ймовірністю q успіху. Ймовірність для даного значення k дорівнює

$$P(k)=\binom{k+z-1}{k}p^z q^k, \tag{22}$$

де $\binom{n}{k}=\frac{n!}{(n-k)!k!}$ – біноміальний коефіцієнт.

Як тільки в чесній мережі буде знайдено z блоків, протягом періоду часу, в ході якого атакуючий сформує $k+1$ блоків (*ми припускаємо, що один блок був попередньо здобутий атакуючим до початку атаки*), гонка починається з відставанням в $z-k-1$ блоків. Звідси випливає, що ймовірність подвійної витрати, коли продавець очікує z підтверджень, дорівнює (23)

$$r=\sum_{k=0}^{\infty}P(k)a_{z-k-1}=\sum_{k=0}^{z-1}\binom{k+z-1}{k}p^z q^k\left(\min\left(q/p,1\right)\right)^{z-k}+$$

$$+\sum_{k=z}^{\infty}\binom{k+z-1}{k}p^z q^k=\begin{cases}1-\sum_{k=0}^{z}\binom{k+z-1}{k}\left(p^z q^k-p^k q^z\right), & \text{якщо } q<p;\\ 1, & \text{якщо } q\geq p.\end{cases} \tag{23}$$

».

Зауважимо, що тут М. Розенфельд робить деякі припущення, що і С. Накамото (*Припущення С. Накамото №№ 1, 2, 3, 4, 5, 8*), але виправляє помилку, що пов'язана з *Припущеннями С. Накамото № 6 та № 9*.

У ряді робіт [16,17] їх авторами звернено увагу на те, що потрібно враховувати час синхронізації у мережі. У цих роботах представлений перший аналіз вразливості системи Біткойн з мережевої точки зору, крім того, ґрунтуючись на експериментальних даних, стверджується, що Біткойн є сильно централізований. Так, як наведено в [18], функція розподілу суми однаково розподілених експоненціальних величин є розподіл Ерланга.

Таким чином, можливо додатково виділити *Припущення № 10* (*яке присутнє у розглянутих розрахунках С. Накамото та М. Розенфельд*), сутність якого полягає в наступному: – результати отримані в припущенні про те, що час поширення блоку в мережі дорівнює нулю.

5 Висновки

В представленій роботі запропоновано критичний аналіз відомих публікацій щодо оцінці ймовірностей подвійний витрати в протоколі консенсусу *«Доказу виконаної роботи»*. Продемонстровано наявність невідповідності та необґрунтованих припущень у декількох відомих роботах, наприклад, таких як роботи Сатоши Накамото [1] та Мені Розенфельда [15].

Виконано аналіз моделей на підставі яких С. Накамото та М. Розенфельдом (*а також іншими авторами, що проводили уточнення отриманих виразів*) були зроблені спроби отримати кількісну оцінку ймовірності успішної атаки подвійний витрати на деякі алгоритми консенсусу, що мають ймовірнісну завершеність. Наведені припущення (*спрощення та допущення*), що мають місце у моделях, та за допомогою яких було отримано кінцевий вираз.

До основних припущень, які були зроблені в роботах, що розглядаються, слід віднести:

Припущення № 1 (С. Накамото, М. Розенфельд) – гонку між чесними учасниками і нападаючим можна уявити, як біноміальне випадкове блукання;

Припущення № 2 (С. Накамото, М. Розенфельд) – ймовірність перемоги зловмисника еквівалентна задачі про «розорення гравця»;

Припущення № 3 (С. Накамото, М. Розенфельд) – зловмисник має можливість (бажання) нескінченно довго проводити атаку на мережу, формуючи альтернативний ланцюг, тобто зловмисник має необмежені для цього ресурси;

Припущення № 4 (С. Накамото, М. Розенфельд) – ймовірності формування блоку чесною мережею або зловмисником вважаються константами, що не змінюються у часі;

Припущення № 5 (С. Накамото, М. Розенфельд) – одержувач замовлених товарів створює новий гаманець (*пару ключів*) безпосередньо перед підписанням транзакції.

Припущення № 6 (С. Накамото) – функція прогресу зловмисника відповідає розподілу Пуассона;

Припущення № 7 (С. Накамото, М. Розенфельд) – група подій в гонці між чесною мережею та зловмисником складається тільки з двох подій, ймовірності яких однозначно пов'язаним між собою співвідношенням $p + q = 1$;

Припущення № 8 (С. Накамото, М. Розенфельд) – вираз, щодо ймовірності успішного формування зловмисником свого альтернативного ланцюжка, отримано для випадку, коли зловмисник лише наздожене, а не випередить, чесну мережу.

Припущення № 9 (С. Накамото) – випадкову величину кількості сформованих блоків було замінено на математичне очікування цієї величини.

Припущення № 10 – результати були отримані в припущенні, що час поширення блоку в мережі дорівнює нулю.

В наведеної роботі авторським колективом розглянута задача розорення гравця та її невелика варіація, сутність якої описано в п. 2.4. Підкреслено існування аналогії з атакою «подвійної витрати на блокчейн» та проведено «Експеримент Пуассона». Розглянуто вірогідність випередження зловмисником чесної мережі в кількості сформованих блоків у загальному випадку.

Посилання

[1] Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System / Satoshi Nakamoto., 2009. – 9 с.

[2] Hackernoon: Two Ways to Double-Spend https://medium.com/hackernoon/bitcoin-core-bug-cve-2018-17144-an-analysis-f80d9d373362 (дата звернення 28.12.19)

[3] BitcoinCore: CVE-2018-17144 Full Disclosure https://web.archive.org/web/20191005023317/https://bitcoincore.org/en/2018/09/20/notice/ (дата звернення 28.12.19)

[4] A. Pinar Ozisik., Brian Neil Levine. An Explanation of Nakamoto's Analysis of Double-spend Attacks https://arxiv.org/pdf/1701.03977.pdf (дата звернення 28.12.19)

[5] W. Feller. An Introduction to Probability Theory and its Applications: Volume I, Volume 3. John Wiley & Sons London-New York-Sydney-Toronto, 1968.

[6] Pinzón C., Rocha C. Double-spend Attack Models with Time Advantange for Bitcoin. Electronic Notes in Theoretical Computer Science. Volume 329, 9 December 2016, Pages 79-103 https://doi.org/10.1016/j.entcs.2016.12.006 (дата звернення 28.12.19)

[7] Rosenfeld M. Analysis of hashrate-based double-spending / Meni Rosenfeld., 2014. – 13 с (arXiv preprint arXiv:1402.2009)

[8] Ozisik A. P. and Levine B. N., “An explanation of Nakamoto’s analysis of double-spend attacks,” arXiv preprint arXiv:1701.03977, 2017 https://arxiv.org/pdf/1701.03977.pdf (дата звернення 30.12.19)

[9] Zaghloul, E., Li, T., Mutka, M.W., & Ren, J. (2019). Bitcoin and Blockchain: Security and Privacy. ArXiv, abs/1904.11435

[10] A.W.F. Edwards. Pascal's problem: The «gambler's ruin». Revue Internationale de Statistique, 51(1):73-79 (http://www.jstor.org/stable/1402732), Apr 1983

[11] Ковальчук Л.В. Основні визначення у галузі блокчейну та детальний аналіз результатів Накамото-Розенфельда-Грунспана про ймовірність атаки подвійної витрати. Звіт про НДР (проміжний), Харків, АТ ІІТ, 36 с.

[12] L. Rey-Bellet. Gambler's ruin and bold play. http://people.math.umass.edu/~lr7q/ps_files/teaching/math456/Week4.pdf, June 7 201651% Attack Explained: The Attack on a Blockchain https://www.fxempire.com/education/article/51-attack-explained-the-attack-on-a-blockchain-513887 (дата звернення 30.12.19)

[13] Walpole R. E., Myers R. H., Myers S. L., Ye K. Probability & Statistics for Engineers & Scientists. Prentice Hall, (See pg. 161 for a discussion of Poisson experiments), 9-th edition, 2012.
[14] Grunspan C., Pérez-Marco R. Double spend races. 2017. hal-01456773 https://hal.archives-ouvertes.fr/hal-01456773 (дата звернення 11.01.20)
[15] Rosenfeld M. Analysis of hashrate-based double-spending / Meni Rosenfeld., 2014. – 13 c (arXiv preprint arXiv:1402.2009)
[16] Apostolaki M. Hijacking Bitcoin: Routing Attacks on Cryptocurrencies / M. Apostolaki, A. Zohar, L. Vanbever. – San Jose, CA, USA, 2017. – 18 p. https://btc-hijack.ethz.ch/files/btc_hijack.pdf
[17] Apostolaki M., Marti G., Müller J., Vanbever L. SABRE: Protecting Bitcoin against Routing Attacks. –San Diego, CA, USA, 2019. pp. 1-15 https://dx.doi.org/10.14722/ndss.2019.23252
[18] Kaidalov D.S., Kovalchuk L.V., Nastenko A.O., Rodinko M.Yu., Shevtsov O.V., Oliynykov R.V. Comparison of block expectation time for various consensus algorithms. Radio Electronics, Computer Science, Control. 2018. № 4. pp. 159-171 DOI 10.15588/1607-3274-2018-4-15

Рецензент: Владимир Хома, д.т.н., проф., Опольский Политехнический Университет, Ополе, Польша.
E-mail: kalash@itesm.mx

Поступила: Декабрь 2019.

Авторы:
Владислав Сафоненко, студент ФКН, ХНУ имени В. Н. Каразина, Харьков, Украина. E-mail:vladyslavsafonenko@gmail.com
Никита Гончаров, студент ФКН, ХНУ имени В. Н. Каразина, Харьков, Украина. E-mail: wdpgames@yandex.ru
Сергей Даценко, студент ФКН, ХНУ имени В. Н. Каразина, Харьков, Украина. E-mail: sergdacenko@gmail.com
Елизавета Лазарева, студентка ФКН, ХНУ имени В. Н. Каразина, Харьков, Украина. E-mail:lazareva15elizaveta@gmail.com
Николай Полуяненко, к.т.н., доцент кафедры, ХНУ имени В. Н. Каразина, Харьков, Украина. E-mail: nlfsr01@gmail.com

Расчет вероятности атаки разветвления блокчейн реестра.

Аннотация. В работе систематизированы сведения по тематике вопроса атаки разветвления блокчейн реестра. Предложен обзор и обобщение информации, представленной в наиболее авторитетных работах в данном направлении. Осуществлен анализ соответствующих работ по оценке вероятности двойной траты в протоколе консенсуса "Доказательства выполненной работы". Рассмотрена проблематика разорения игрока и проведена аналогия с атакой двойной траты на блокчейн. Рассмотрен эксперимент Пуассона для общего случая. Проанализированы модели, на основании которых, С. Накамото и М. Розенфельдом были предприняты попытки получить количественную оценку вероятности успешной атаки двойной траты на некоторые алгоритмы консенсуса, имеющие вероятностную завершенность. Приведены упрощения и допущения, имеющие место в соответствующих моделях, с помощью которых получено конечное выражение.

Ключевые слова: компьютерные сети; децентрализованные системы; блокчейн технологии; атака на блокчейн; атака разветвления; атака двойной траты; эксперимент Пуассона.

Reviewer: Volodymyr Khoma, Dr. of Sciences (Eng.), Full Prof., The Opole University of Technology, Opole, Poland.
E-mail: kalash@itesm.mx

Received: December 2019.

Authors:
Vladyslav Safonenko, CSD Student, V.N. Karazin Kharkiv National University, Ukraine. E-mail: vladyslavsafonenko@gmail.com
Nikita Goncharov, CSD Student, V.N. Karazin Kharkiv National University, Ukraine. E-mail: wdpgames@yandex.ru
Sergey Datsenko, CSD Student, V.N. Karazin Kharkiv National University, Ukraine. E-mail: sergdacenko@gmail.com
Elizaveta Lazareva, CSD Student, V.N. Karazin Kharkiv National University, Ukraine. E-mail: lazareva15elizaveta@gmail.com
Nikolay Poluyanenko, Ph.D., Associate Prof., V.N. Karazin Kharkiv National University, Ukraine. E-mail: nlfsr01@gmail.com

Alternative history attack success probability calculation in blockchain system.

Annotation. This article systemizes the information on the subject of the alternative history attack of the blockchain registry. The review and generalization of the information presented in the most respected works in this direction is offered. The analysis of corresponding works on estimation of probability of double spending in the "Proof of Work" consensus protocol is carried out. The problems of the player's ruin are considered and an analogy with the attack of double spending on the blockchain is made. Poisson's experiment for the general case is considered. The models on the basis of which S. Nakamoto and M. Rosenfeld made attempts to get a quantitative estimation of probability of successful double spending attack on some algorithms of consensus having probability completeness are analyzed. Simplifications and assumptions that take place in the respective models with the help of which the final expression is obtained are given.

Keywords: Computer networks; Decentralized systems; Blockchain technology; Alternative history attack; Double spending attack; Poisson's experiment.

УДК 004.056.5 DOI: 10.26565/2519-2310-2019-4-03

ОСОБЛИВОСТІ ЗАХИСТУ КОРПОРАТИВНИХ РЕСУРСІВ ЗА ДОПОМОГОЮ ТЕХНОЛОГІЇ HONEYPOT

Сабіна Рузудженк, Каріна Погоріла, Тетяна Кохановська, Сергій Малахов

Харківський національний університет імені В.Н. Каразіна, майдан Свободи 4, Харків, 61022, Україна
ruzudzhenk.jb@gmail.com, karina.pogorelka@gmail.com, tanya.koh99@gmail.com, mailgate@meta.ua

Рецензент: Олександр Оксіюк, д.т.н., проф., Київський національний університет імені Т. Шевченка, вул. М. Ломоносова 81, Київ, 03189, Україна.
o.oksiuk@gmail.com

Поступила: Листопад 2019.

Анотація: *У статті надано стислий огляд основних можливостей технології Honeypot. Розглянуті питання стосовно: - особливостей моніторингу мережевої активності на різних етапах розвитку атак/вторгнень; - розміщення датчиків системи; - процедур збору та узагальнення даних щодо мережевих подій; - варіантів модифікації інструментів захисту; - організації структури захисту тощо. Розглянуті загальні принципи роботи відповідних систем на базі відокремлених серверів та програмно емульованих мереж. Узагальнено основні недоліки даної технології. Звернено увагу на перспективність використання різноманітних рішень Honeypot для цілей розширення потенціалу вже розгорнутих засобів забезпечення інформаційної безпеки (ІБ).*

Ключові слова*: Honeypot; вторгнення; інформаційна безпека; ЛОМ; Firewall; IDS; IPS.*

1 Вступ

У сучасному світі інформаційних технологій кожен Інтернет користувач та локальна обчислювальна мережа, які мають підключення до Інтернет, рано чи пізно, але в будь-якому випадку, неминуче стикаються з проблемою зовнішніх кібератак. Враховуючи цей факт, як аксіому, можна зробити висновок, що відповідні інформаційні і апаратні ресурси Інтернет користувачів та локальних обчислювальних мереж (ЛОМ), що взаємодіють з Інтернет, обов'язково повинні бути спроможні парирувати відповідні загрози. Саме тому, на постійній основі, необхідно забезпечувати комплексний моніторинг поточної мережевої активності, особливо в частині аналізу змісту, характеру та інтенсивності трансграничного трафіку. В першу чергу це стосується аналізу трафіку в межах спеціально передбачених демілітаризованих зон, або відповідних публічних сервісів (*при їх наявності*), що передбачають інтенсивну взаємодію з користувачами, які знаходяться за рамками організованого периметра безпеки компанії або окремого користувача. Одним з ефективних засобів ведення моніторингу мережевої активності та виявлення ознак підготовки майбутнього кіберзлочини, є використання можливостей технології Honeypot (*т.з. вузлів або мереж пасток/приманок*). Крім того потенціал Honeypot дозволяє, в буквальному сенсі, виграти час, відволікаючи мережевого зловмисника або спеціалізовану програму на виконання завідомо зайвих або хибних дій.

Аналіз джерел. Honeypot вперше з'явилися з першими комп'ютерними зловмисниками, а практика їх активного використання налічує вже більше 20 років. Роботи по їх створенню та практичному впровадженню проводилися паралельно з дослідженнями IDS та IPS [1]. Першою документальною згадкою за тематикою Honeypot, була робота Кліффорда Столла «The Cuckoo's Egg», що вийшла у 1990 році. А вже у 2000-х роках Honeypot стали досить поширеними системами, що забезпечували ефективну протидію спробам несанкціонованого проникнення до «внутрішнього» периметру безпеки комп'ютерних мереж компаній.

На сьогоднішній день актуальним напрямом використання вузлів і мереж – приманок є, наприклад, протистояння діям кіберзлочинців при парируванні розгалужених атак типу «відмова в обслуговуванні (*DDos*) при використанні різних моделей хмарних обчислень (*наприклад, PAAS або IAAS*). Застосування Honeypot полегшує збір інформації про потенційну атаку і атакуючого вже на етапах підготовки (*попередньої розвідки*) та початку «проникнення» в

відповідну систему. В цілому, Honeypot можна використовувати, як дуже ефективне доповнення до технологій виявлення та запобігання несанкціонованих вторгнень (IDS/IPS) [2,3].

Як правило, створюючи Honeypot (далі HPot) фахівці з інформаційної безпеки (ІБ) очікують від нього/неї вирішення 3-х основних завдань: 1 - отримання змістовної інформації щодо використовуваних кіберзлочинцями способів та методик проникнення до ресурсів мережі, що захищається; 2 - реєстрація моменту початку атаки (*несанкціонованого проникнення до мережі або доступу до вузла-пастки*); 3 - забезпечення виграшу часу, за рахунок переорієнтації уваги хакера з фактичних елементів і ресурсів ЛОМ на їх неіснуючі мережеві клони (*копії-пастки*). В цілому HPot, це досить гнучкий інструмент, який, залежно від умов роботи (*шлюз, вузол, сукупність вузлів, гібрид*) та кінцевих завдань (*захист від спам-ботів або система раннього попередження про мережеві інциденти тощо*), можливо застосовувати в різних іпостасях [4]. Так наприклад, це може бути програмний «емулятор» іншої системи, додаток або стандартна система/системи. Як свідчить досвід [1-2, 4-5], HPot може виконувати досить різні завдання, наприклад: - визначати початок атаки; - збирати інформацію про несанкціонований моніторинг поточної мережевої активності; - фіксувати інформацію про задіяні механізми витоку даних, служити засобом захисту від спам-ботів тощо. При цьому впровадження HPot може переслідувати діаметрально протилежні цілі, наприклад: організовувати виробничі і/або дослідні пастки, де перші зорієнтовані на питаннях захисту мережі, а другі на зборі відповідних додаткових уточнюючих відомостей [6].

Актуальність. Таким чином, розгляд питань стосовно особливостей використання HPot в сучасних умовах та можливостей їх подальшої модифікації в майбутньому (*віртуалізація процедур аналізу трафіку в визначених мережевих сегментах, адаптивне шлюзування мережевого трафіку, оперативна кластеризація HPot, врахування можливостей блокчейн тощо*), не визиває ніяких сумнівів.

2 Основна частина

Зазвичай HPot являє собою програмно-апаратний комплекс, якій складається з наступних основних компонентів: - вузол-приманка, мережевий сенсор і накопичувач даних про всю аномальну мережеву активність [7]. В якості майбутньої приманки організовується певний сервер (Рис. 1), що працює під управлінням довільної операційної системи (ОС), та настроюється на потрібний, в даних умовах, рівень безпеки (*протидії потенційному нападнику*). Ізольованість від інших ділянок (*сегментів*) ЛОМ, потенційно перешкоджає використанню вузла-приманки, як платформи для майбутніх атак на інші елементи мережі, однак, надає хакеру можливість швидко зрозуміти, що він вже на півдорозі до «успіху». Найчастіше вузлів з приманкою буває кілька. В цьому разі одні з них розраховані для протидії хакерам-початківцям, а інші – більш захищені та «тонко» налаштовані, орієнтовані на виявлення ще невідомих технік злому, які очікуються з боку більш досвідчених мережевих зловмисників.

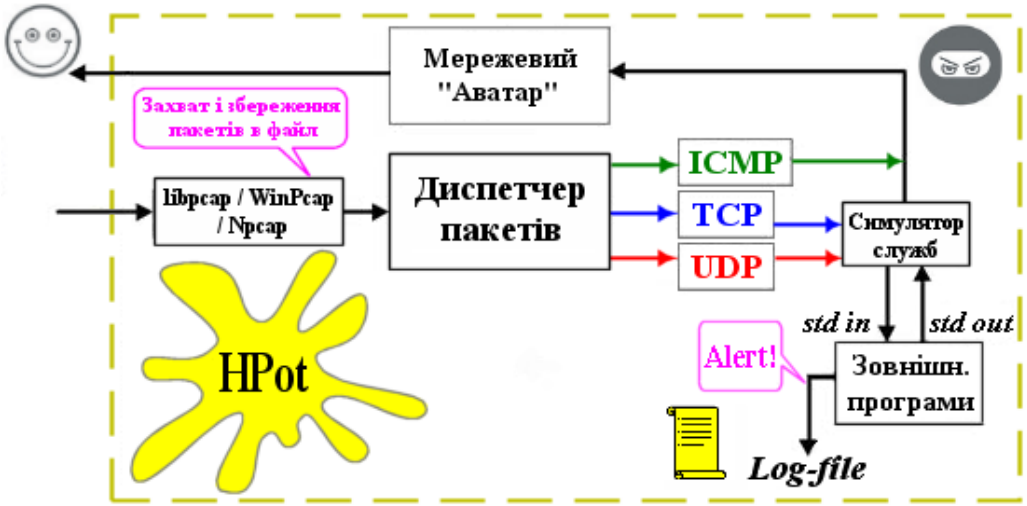


Рис. 1 – Спрощена схема Honeypot з низьким рівнем взаємодії *(варіант)*

Мережевий сенсор найчастіше реалізується на базі UNIX-подібних операційних систем (ОС), а для моніторингу поточної інформації використовується утиліта *tcpdump* або її аналоги [2]. При цьому, залежно від конфігурації мережі, що захищається, сенсор може знаходитись на одному з вузлів визначеного сегмента ЛОМ, або бути маршрутизатором, якій розташований перед приманкою. Іноді мережевий сенсор поєднується безпосередньо з самої приманкою, що істотно спрощує та здешевлює реалізацію HPot, однак, послаблює безпеку всієї системи (*захопивши управління приманкою, атакуючий швидко виявить сенсор, та нівелює всю систему*). Розміщення сенсора на одному з вузлів визначеного сегмента мережі забезпечує йому найбільшу конфіденційність *(мають на увазі труднощі з його виявлення)*. При цьому мережевий інтерфейс сенсора може і не мати власної IP-адреси, прослуховуючи трафік в т.з. «стелс-режимі». При виносі сенсору системи на маршрутизатор *(перед приманкою),* визначити чи працює на ньому мережевий сенсор, чи ні, в загальному випадку є дуже складно.

Іншим напрямом є використання програмно емульованих HPot [7]. На відміну від фізично розгорнутих на окремому сервері/вузлі систем, програмно емульовані рішення швидко відновлюються в разі їх злому, а також досить чітко обмежуються (*відокремлюються*) від «основної» ОС. Подібні рішення можуть бути створені, наприклад, за допомогою віртуальної машини або Honeyd [5].

Основна відмінність між ними полягає у різниці масштабів корпоративної обчислювальної мережі, що потребує захисту. Так наприклад, у випадку малої офісної ЛОМ не має особливого сенсу організовувати виділений сервер для цілей протоколювання підозрілих мережевих подій. В цьому разі досить буде обмежитися віртуальною системою або навіть одним віртуальним сервісом. Однак, у великих організаціях з розвиненою IT-структурою, потрібно використовувати саме виділені сервери з повністю відтвореними на них мережевими службами. При цьому, зазвичай в конфігурації таких служб навмисно допускають відповідні помилки-пастки, щоб у потенційного зловмисника обов'язково «вдався» злом системи. А як було зазначено вище, саме у цьому і полягає основна мета впровадження Honeypot – привернути увагу зловмисника та виграти у нього час, для організації ефективного парирування даного різновиду загроз ІБ.

Слід зазначити, що залежно від ступеня взаємодії зі зловмисником HPot діляться на 3 типи: - слабкої, середньої та сильної взаємодії. Основна відмінність відповідних реалізацій систем полягає в складності їх розгортання, використання і підтримки, а також в забезпечуваних рівнях імітації та протоколювання даних мережевої активності. Так, засоби слабкої взаємодії порівняно легко розгортаються та більш прості у подальшої експлуатації, але вони і менш ефективні (*с точки зору можливостей реєстрації інформації та ступеню мережевої мімікрії*). В цьому сенсі засоби систем з сильною взаємодією забезпечують більш глибокій рівень протоколювання і імітації, проте вони є і більш складними у використанні, та потребують більш високих професійних компетенцій персоналу. Крім того, впровадження такого типу HPot тягне за собою більш високий ризик їх виявлення і наступної компрометації.

В цілому, незважаючи на те, що HPot ретельно налаштовані (*в наслідок чого їх важко виявити*), при певних зусиллях та обачливості можливо виявити деякі ознаки того, що ми маємо справу саме з ними. Так наприклад, у разі використання фішингу [2], зловмисні URL–адреси часто відрізняються від легітимних ресурсів тільки однією літерою або цифрою, тому відповідна пильність зовсім не є надмірною. В цьому разі, перш за все, потрібно завжди уважно перевіряти правильність вводу відповідної URL-адреси. Крім цього, якщо «довірений» веб-сайт раптом починає запитувати облікові дані або будь-яку іншу «чутливу» інформацію, чого раніше за ним помічено не було, це також повинно насторожити.

На практиці існують достатньо прості кроки, що дозволяють уникнути можливих наслідків: - ніколи не клікати підозрілі посилання, що отримані з неперевірених або невідомих для користувача джерел; - отримане посилання більш безпечно перенабрати в адресної строчці браузеру, ніж клікати отримане посилання; - використовувати безпечні DNS сервіси [8]; - перевірити наявність сайту в переліку DBL (*реєстрі заблокованих доменів*) [9].

У разі проведення атаки, хакеру важливо мати надійні та швидкі канали зв'язку, щоб забезпечити собі можливості маневру проти спроб його відстеження. Так наприклад, в разі перебування в широкомовній мережі, для успішного маскування дій він може обмежитися клонуванням чужих IP і/або MAC-адрес [2, 9]. В цьому випадку, за умови, що ЛОМ яка атакується не має ніякого додатково обладнання, для визначення дій порушника, ідентифікувати зловмисника буде практично неможливо, хоча і тут є одне «але». Так, в разі якщо комп'ютер хакера вразлив, то HPot може непомітно для нього розмістити і активувати відповідного програмного "жучка" з усіма можливими наслідками *(наприклад cookie, що передані через браузер)*. Крім того, очікуючи атаку зловмисника, треба враховувати, що при її здійсненні він може передбачити певні захисні заходи, наприклад: використовувати ланцюжок з кількох комп'ютерів або гаджетів, а в глобальну мережу виходити з однієї з публічних точок доступу, якнайдалі від свого основного місця розташування, щоб значно ускладнити свою локацію. В любому випадку виходити в мережу по комутованому доступу для нього буде вкрай небезпечно, та скоріш за все буде впроваджено механізм каскадних *proxy*. Але і в цьому разі йому не слід цілком покладатися на можливості *proxy*, оскільки заздалегідь ніколи невідомо фактичний рівень протоколювання даних підключення на кожному з задіяних *proxy*-серверів. В цьому контексті слід зауважити, що частина безкоштовних *proxy* в дійсності є своєрідними приманками, що встановлені та підтримуються відповідними службами.

Враховуючи специфіку технології HPot для забезпечення належного рівня захисту корпоративних ресурсів (*перш за все інформаційних*) слід враховувати можливі способи обходу потенційними зловмисниками цих приманок. Так, серед іншого, з метою перешкоджання спроб їх ідентифікації, зловмисники можуть запровадити наступні дії:

- використовувати технологій анонімізації і тунелювання;
- впровадити технології мережевого аватару *(вигаданої віртуальної особистості, що ніяк не пов'язана з реальною)*;
- відмовитись або значно обмежити чисельність завантажень будь-яких файлів (в тому числі ПЗ), окрім випадків, коли вони будуть отримані з вкрай надійного, з їх точки зору, джерела.

В даний час окрім відповідних рішень технології HPot та її похідних (*Honeynet, Honeytoken, Honeyd тощо*) в різних комбінаціях активно застосовуються і інші засоби забезпечення ІБ комп'ютерних мереж такі, як Padded Cell, міжмережеві екрани, IDS, IPS, DLP тощо [2-4, 8, 10]. Останні рішення являють собою не пасивні приманки, а активно протидіючи засоби парирування, як відомих, так і ще невідомих загроз ІБ. Найбільш близьким до HPot за низькою функціональних ознак є IDS (*система виявлення вторгнень*), що протоколює всі «зовнішні» мережеві підключення до системи, в цілому, або її окремого сегменту/сегментів. Honeynet поєднує у собі декілька Honeypot-приманок, що поєднані єдиним задумом, та складають інтегровану мережу-пастку. Honeytoken є приманкою, основним завданням котрої є виявлення випадків неправомірного використання даних. В основу роботи Honeytoken положено принцип «попередити, а не запобігти». На відміну від них, Padded Cell – це своєрідний різновид приманки типу «пісочниця». Потрапляючи до неї, зловмисник істотно обмежений в можливостях завдати шкоди системі-жертві, так як він фактично розташований в ізольованому від іншої системи/процесів середовищі. На відміну від попередніх, DLP-системи, є більш інтегровані та високоінтелектуальні системи, які поєднують єдиним задумом (алгоритмом) роботу багатьох інших підсистем та засобів «активного» та «пасивного» мережевого захисту (*ті ж самі HPot, firewall. IDS та IPS*).

В цілому, пам'ятаючи про основне призначення мережевих приманок – створення максимальної ілюзії фактичного доступу до «внутрішніх» корпоративних ресурсів, приманка повинна бути спеціальним чином підготовлена. Іншими словами, вона повинна містити фальшиву інформацію, яка своїм складом і контекстом, відповідає основному профілю діяльності компанії-жертви. В такому випадку у потенційного зломщика системи, буде підтримуватися стійке враження про успішне завершення реалізованої їм атаки. При менш оптимістичному для зломщика результаті подій, використання HPot забезпечить виграш часу для сторони, що

захищається (*фахівців компанії, які відповідають за питання ІБ*), таким чином, даючи їм можливість модифікувати тактику захисту, та «втягнути» хакера в більш тривалий цикл мережевого протистояння. В принципі ця «гра», якщо і не відіб'є у хакера бажання «зламати» обрану їм мережу, то, принаймні, тільки посилить у нього враження про реальність того, що відбувається, що і потрібно отримати, в кінцевому підсумку.

3 Основні переваги та недоліки технології Honeypot

Аналіз особливостей захисту корпоративних ресурсів за допомогою використання відповідних мереж приманок, дозволяє виділити, як ряд очевидних переваг, так низку специфічних недоліків даної технології [11,12]. Так, до сильних сторін HPot слід віднести наступне:

1. *Ретельний збір змістовної інформації про мережеві події*. Засоби HPot збирають невелику кількість даних (*порядку кілька мегабайт на добу*), але, зазвичай, вони мають принципове, з точки зору аудиту ІБ, значення. Дані, які були зареєстровані HPot, найбільш ймовірно є результатами сканування, несанкціонованого дослідження (мережевою розвідкою) або ознаками атаки - тобто є інформацією, що має високий пріоритет для відповідного аналітика. Таким чином, HPot надає аудитору ІБ практично всю потрібну інформацію, причому фактично в режимі реального часу. Це спрощує аналіз мережевої активності, зменшує час реакції, та надає можливість вжити превентивні заходи щодо завчасного парирування відповідних загроз ІБ.

2. *Невимогливість до системних ресурсів*. З точки зору питань підтримки безперервності процесів забезпечення ІБ, нестача ресурсів – це ситуація, коли задіяні механізми HPot не можуть бути продовжені, тому що наявні або виділені для цього ресурси вже вичерпані. Але, в цілому, використання HPot забезпечує потрібний баланс завдань ІБ та наявних ресурсів.

Оскільки засоби Honeypot, в своєї переважної більшості, спрямовані на завдання контролю і фіксації мережевої активності в визначеному сегменті/вузлі ЛОМ, то зрозуміло, що вони зазвичай не схильні до проблем нестачі ресурсів. Це відрізняє HPot від більшості IDS (систем виявлення вторгнень), які мають певні складності при забезпеченні завдань ІБ в високошвидкісних ІТ-структурах.

3. *Очевидність практичного використання*. Існуючи засоби HPot оперативно та неодноразово підтверджують своє основне призначення, як інструменту перманентної мережевої розвідки, всякий раз, коли їх елементи піддаються нападу або неавторизованому зондуванню, підтверджуючи тім самим прояви несанкціонованої мережевої діяльності.

У будь-якому випадку HPot є досить простим і переконливим засобом підтвердження правоти твердження: - що, якщо ви параноїк, то це ще не означає, що за вашої ІТ-структурою ніхто не спостерігає. Таким чином, коли хтось раптом вирішить, що вже немає ніяких загроз, то саме HPot зможе ефективно довести зворотне та документально підтвердити, що загроза бути скомпрометованим постійно була поряд, і більш того, очікує саме на вас.

4. *Простота розгортання та подальшої експлуатації*. Порівняльна простота процедур встановлення та первинного конфігурування HPot є найбільш вагомим аргументом на користь даної технології. Для більшої об'єктивності слід зазначити, що для різних HPot існують і різні додаткові функціональні розширення *(бази сигнатур відомих атак, бази типових реакцій і т.ін.)*, але в любому випадку *(в незалежності від типу системи та місця її розгортання)* фундаментальна парадигма залишається незмінною: – якщо хтось або щось з'єднується з HPot, то це вимагає обов’язкової перевірки.

Як вже було зазначено вище, поряд з очевидними та вкрай корисними можливостями HPot, вони мають і ряд характерних недоліків, до яких слід віднести:

1. *Можливість «розкриття» (демаскування) пастки*. Можливість розкриття HPot зловмисником – в любому випадку базується на його компетенціях зі збору та узагальнення відповідної інформації, за наслідками котрої він саме і може ідентифікувати істинну сутність досліджуємого об’єкту (*за сукупністю очікуваних характеристик або особливостей мережевої поведінки*). В загальному випадку можливість компрометації HPot залежить від двох основних факторів: - кваліфікації зловмисника та коректності налаштувань самого HPot. Тому,

якщо питання селекції рівня професійної кваліфікації зловмисника виходить за рамки компетенції відповідних фахівців з питань корпоративної ІБ, то питання конфігурування та впровадження коректних налаштувань HPot, повністю залежить від відповідного персоналу.

2. *Обмежену область спостереження (контролю)*. Від самого початку Honeypot здійснюють моніторинг мережевої діяльності, яка була спрямована саме проти них. В разі, якщо дії атакуючого будуть спрямовані на інші елементи ЛОМ, HPot практично вже не буде спроможним фіксувати дану діяльність. В враховуючі цю особливість, слід постійно мати на увазі, що в разі ідентифікації зловмисником HPot (*безвідносно того, як саме він це зробив*) він може спробувати «обійти» його, та переключити свою увагу на інші – справжні елементи мережі, що захищається. В цьому випадку, для фахівців які забезпечують захист мережі, фактор часу стає критичним, тому що компрометована пастка в буквальному сенсі «вимкнена з гри» (*аналогічно ситуації коли противнику відома карта міного поля, що дає йому можливість прокласти безпечний маршрут*). Таким чином, обмежена зона контролю кожного окремого HPot, практично виключає контроль мережевих подій поза зоною його відповідальності. Як наслідок, для подолання цього ефекту необхідно комбінувати порядок розміщення датчиків HPot, тобто: - або дублювати, або каскадувати відповідні елементи HPot, що призводить до загального ускладнення відповідної системи та зростанню її вартості.

3. *Збільшення ймовірності початку таргетованих атак елементів інших IT-структур, внаслідок випадків компрометації/злому кожного HPot*. Різні HPot мають різні рівні ризику, для елементів інших мереж. Так деякі з них мають дуже невеликий ризик, в той час, як інші надають атакуючому досить широкі можливості для наступних атак (*наприклад, створення фішингових ресурсів або бот-систем для спамерів*). В цьому сенсі, існує проста думка, що чим простіше діючий HPot, то тим менший потенційний ризик його подальшого протиправного застосування.

4 Висновки

1. Основні переваги HPot, серед іншого, полягають в їх гнучкості та масштабованості. На даний час у мережевих злочинців поки все ще немає досконалих методик детектування та подолання захисних механізмів різних HPot, проте, стратегії мережевої розвідки і методи атак постійно прогресують, тому питання побудови ефективної адаптивної протидії новим мережевим загрозам є одним із найважливіших напрямів роботи відповідних фахівців.

2. Архітектура різних HPot, в цілому, достатньо добре відома і тому, потенційно, вразлива. Однак, наділяючи HPot-рішення більш складним сценарним контекстом та скорочуючи час мережевої експозиції можливо підтримувати їх потенціал в досить паритетному стані. Обидва напряму потребують більш щільної уваги відповідних фахівців (*аналіз лог-файлів і вдосконалення алгоритмів роботи «мережевого аватару» HPot*), та підтримки їх професійного реноме на досить високому рівні (*постійне навчання та вдосконалення навичок*).

3. В контексті сказаного, з великою часткою впевненості можна стверджувати, що систематизація правил роботи мережевого аватару HPot (*як сукупності користувальницьких поведінкових алгоритмів*) та синтез уніфікованих наборів відповідних поведінкових профілів для HPot, бачиться, як завдання, що важко формалізується (*в першу чергу, через різноманіття варіантів мережевої активності, що притаманна для кожної конкретної реалізації окремих мереж та визначених мережевих вузлів*).

4. Надлишкова уніфікація поведінкових профілів HPot певною мірою може полегшити потенційному мережевому зловмиснику процес моніторингу і наступної ідентифікації HPot (*за сукупністю характерних ознак*), тому формування пулу відповідних мережевих аватарів слід розглядати, не більше, як основу для її подальшої адресної підгонки (*налаштування*) під специфіку характерних завдань, топологічні та інші особливості кожної IT-структури або їх окремих елементів (вузлів).

5. HPot не підміняють собою інших механізмів безпеки, а лише ефективно розширюють наявний арсенал засобів мережевого моніторингу, та в певній мірі, протидії новим загрозам безпеки (*перш за все, як інструменту швидкого або випереджаючого реагування*). Тому

шлях інтеграції HPot з іншими, вже розгорнутими рішеннями ІБ, є найбільш збалансованим напрямом подальшого підвищення загального рівня безпеки мережевих ресурсів.

6. Honeypot є гнучким та достатньо бюджетним інструментом отримання первинної інформації для дослідження нових методик і різновидів дій мережевих зловмисників. Ретельний аналіз відповідної інформації дозволяє досить ефективно відслідковувати, як еволюцію технік мережевого нападу, так і прийомів маніпулювання мотиваціями кіберзловмисників.

Посилання

[1] Сильнов Д. С., Титов К. Е., Разработка и реализация Honeypot-ловушек сетевых служб, использующих протокол SIP, DOI: https://cyberleninka.ru/article/v/razrabotka-i-realizatsiya-honeypot-lovushki-setevyh-sluzhb-ispolzuyuschih-protokol-sip

[2] Безопасная сеть вашей компании / Джон Маллери, Джейсон Занн и др.; пер. с англ. Е. Линдеманн. – М.: НТ Пресс, 2007. – 640 с.

[3] Ріпний О.С., Дьяченко О.О., Малахов С.В. // Особливості функціонування систем IDS та IPS при реалізації спроб несанкціонованого доступу до корпоративних ресурсів. Матеріали IX міжнародній НТК. 11-12.04.2019. – Х.: НТУ "ХПІ". – 2019. – С.95.

[4] Honeypot success stories Ел.ресурс. – URL: https://www.drupal.org/docs/8/modules/honeypot/honeypot-success-stories

[5] Honeypots – University of Arizona Ел.ресурс. – URL: https://www2.cs.arizona.edu/~collberg/Teaching/466-566/2012/Resources/presentations/2012/topic12-final/report.pdf

[6] HoneyPot для спамеров. Персональный блог Игоря Агурьянова Ел.ресурс. – URL: http://aguryanov.blogspot.com/2014/08/honeypot-for-spammer.html

[7] M. M. Rehman H., Honeypots and Routers Collecting Internet Attacks, 2015.

[8] Валерий Коржов Google Public DNS как средство защиты. Ел.ресурс. – URL: https://www.anti-malware.ru/node/2299 (дата звернення 12.12.2019)

[9] Chris Moore, Detecting ransomware with Honeypot techniques. DOI: https://ieeexplore.ieee.org/abstract/document/7600214

[10] Череватенко Д. Р., Торбеева М. В., Honeypot как средство информационной безопасности, DOI: http://ir.nmu.org.ua/bitstream/handle/123456789/1679/19.pdf?sequence=1&isAllowed=y

[11] Lance Spitzner, Honeypots: tracking hackers. Ел.ресурс. – URL: http://www.it-docs.net/ddata/792.pdf

[12] Технология Honeypot, Часть 1: Назначение Honeypot. DOI: https://www.securitylab.ru/analytics/275420.php

Reviewer: Oleksandr Oksiiuk, Doctor of Sciences (Eng.), Full Prof., Taras Shevchenko National University of Kiev 81 Lomonosova St., Kyiv, 03189, Ukraine.
E-mail: o.oksiuk@gmail.com

Received on November 2019.

Authors:

Sabina Ruzudzhenk, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: ruzudzhenk.jb@gmail.com

Karina Pogorelaya, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: karina.pogorelka@gmail.com

Tetiana Kokhanovska, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: tanya.koh99@gmail.com

Serhii Malakhov, Ph.D., Senior Research, Associate Prof. of the Department, V. N. Karazin Kharkiv National University, Ukraine.
E-mail: mailgate@meta.ua

Features of protecting corporate resources with Honeypot technology.

Abstract. The article provides a brief overview of the main features of Honeypot technology. The questions concerning are considered: - features of monitoring network activity at different stages of attack development; - placement of system sensors; - procedures for collecting and summarizing data on network events; - options for modifying protection tools; - organization of the protection structure, etc. The general principles of operation of the respective systems based on individual servers and software-emulated networks are considered. The main disadvantages of this technology are formulated. Attention is drawn to the prospects of using various Honeypot solutions to expand the potential of already deployed information security tools (IS).

Keywords: Honeypot; Intrusion; Informational security; LAN; Firewall; IDS; IPS.

Рецензент: Александр Оксиюк, д.т.н., проф., Киевский национальный университет имени Т. Шевченко, ул. Ломоносова 81, Киев, 03189 Украина. E-mail: o.oksiuk@gmail.com

Поступила: Ноябрь 2019.

Авторы:

Сабина Рузудженк, студентка факультета компьютерных наук, ХНУ имени В.Н. Каразина, Харьков, Украина.
E-mail: ruzudzhenk.jb@gmail.com

Карина Погорелая, студентка факультета компьютерных наук, ХНУ имени В.Н. Каразина, Харьков, Украина.
E-mail: karina.pogorelka@gmail.com
Татьяна Кохановская, студентка факультета компьютерных наук, ХНУ имени В.Н. Каразина, Харьков, Украина.
E-mail: kate7smith12@gmail.com
Сергей Малахов, к.т.н., с.н.с., доц. кафедры, ХНУ имени В.Н. Каразина, Харьков, Украина.
E-mail: mailgate@meta.ua

Особенности защиты корпоративных ресурсов с помощью технологии Honeypot.

Аннотация. В статье представлен краткий обзор основных возможностей технологии Honeypot. Рассмотрены вопросы касающиеся: - особенностей мониторинга сетевой активности на разных этапах развития атаки/вторжения; - размещения датчиков системы; - процедур сбора и обобщения данных о сетевых событиях; - вариантов модификации инструментов защиты; - организации структуры защиты и т.п. Рассмотрены общие принципы работы соответствующих систем на базе отдельных серверов и программно эмулируемых сетей. Обобщены основные недостатки данной технологии. Обращено внимание на перспективность использования различных решений Honeypot для целей расширения потенциала уже развернутых средств обеспечения информационной безопасности (ИБ).

Ключевые слова: Honeypot; вторжение; информационная безопасность; ЛВС; межсетевой экран; IDS; IPS.

УДК 004.056.5:621.391 DOI: 10.26565/2519-2310-2019-4-04

ФОРМУВАННЯ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ ДЛЯ ПРИХОВУВАННЯ ДАНИХ В ЗОБРАЖЕННЯХ

Євгеній Деменко[1], Олександр Онікійчук[1], Анна Арищенко[1], Людмила Горбачова[1], Олексій Смірнов[2]

[1] - Харківський національний університет імені В.Н. Каразіна, Харків, Україна
[2] - Центральний український НТУ, Кропивницький, Україна
demenjay@gmail.com, onik4524a@gmail.com, annaarischenko@gmail.com, lusyag23@gmail.com, dr.smirnovoa@gmail.com

Рецензент: В'ячеслав Калашников, д.ф.-м.н., проф., Технологічний університет Монтеррея, Монтеррей, 64849, Мексика
kalash@itesm.mx

Надійшло: Січень 2020.

Анотація: *У статті розглядаються способи приховування даних в цифрових зображеннях з використанням псевдовипадкових послідовностей і технології прямого розширення спектра. Пропонується новий спосіб формування псевдовипадкових послідовностей, який враховує статистичні властивості контейнерів-зображень. Це дозволяє домогтися низької кореляції, що забезпечує надійне і безпечне приховування інформації в цифрових зображеннях. Результати експериментальних досліджень показують, що інтенсивність бітових помилок в відновлених повідомленнях істотно знижена. При цьому викривлення контейнерів-зображень залишаються на колишньому рівні.*

Ключові слова: *стеганографія; приховування даних; цифрові зображення; псевдовипадкова послідовність; технологія прямого розширення спектра.*

1 Вступ

Для приховування інформації в цифрових контейнерах-зображеннях в роботах [1-7] запропоновано використовувати псевдовипадкові послідовності та технологію прямого розширення спектра частот. Ця технологія традиційно використовується в системах радіозв'язку з множинним доступом [9-11]. Псевдовипадкові послідовності, які застосовуються, є слабокорельованими один з одним. Вони можуть бути співставлені різними абонентськими каналами і це дозволяє істотно підвищити ємність множинного доступу, що здешевлює послуги зв'язку. Використовуючи кореляційний прийом, можна виправляти помилки, які виникають, тим самим підвищуючи прихованість зв'язку. Крім того, застосування довгих псевдовипадкових послідовностей дозволяє організувати зв'язок при потужності переданих сигналів нижче рівня природних шумів, що забезпечує екологічність зв'язку [9-11].

Всі ці переваги можна використовувати і в інших додатках, наприклад, в цифровій стеганографії. Так в роботах [12-21] пропонується інтерпретувати зображення-контейнери, як шум в каналах зв'язку. Тоді передача даних по каналу зв'язку (КЗ) з шумами представляється як завдання приховування інформаційних повідомлень в зображеннях-контейнерах (*далі контейнерах*). При цьому завадостійкість інтерпретується як стійкість до помилок в відновлених повідомленнях, а абонентська ємність - як пропускна здатність стеганосистеми.

Слід зазначити, що запропоновані в роботах [12-21] методи приховування даних мають певні недоліки. Наприклад, базове припущення про статистичні властивості контейнера часто не виконується. Дійсно, реалістичні зображення володіють високою природною надмірністю, їхні окремі пікселі сильно корельовано між собою, тобто цифрове зображення статистично не подібно до природного шуму в КЗ. Практично це означає можливість кореляції застосовуваних псевдовипадкових послідовностей (ПВП) і контейнерів. У цьому випадку відновлення інформаційних повідомлень на приймальній стороні часто відбувається з великим числом помилок. Наприклад, в роботі [14] показано, що інтенсивність бітових помилок (*Bit Error Rate - BER*) в відновлених повідомленнях від 10% до 30% і ніколи не опускається нижче 10% (*навіть при дуже високій потужності дискретних сигналів - псевдовипадкових послідовностей*). Це змушує використовувати досить складні технології корекції помилок, засновані на внесенні додаткової надмірності [15].

Мета даної роботи полягає в розгляді та дослідженні іншого способу зниження помилок в повідомленнях що відновлюються. Так, замість корекції помилок пропонується використовувати спеціально сформовані ПВП, т.зв. адаптивну генерацію, тобто процедуру, яка при формуванні послідовностей враховує статистичні властивості контейнерів. Проведені експериментальні дослідження підтвердили успішність зазначеного підходу, адже вдається істотно знизити інтенсивність помилок в відновлених повідомленнях (*стеганоконтенті*). При цьому викривлення контейнерів залишаються на колишньому рівні.

2 Технологія прямого розширення спектру в стеганографії

У роботах [12-21] розглянуті базові поняття та методи цифрової стеганографії з використанням технології прямого розширення спектра. Нижче описано процес приховування інформаційних повідомлень та їх відновлення на приймальній стороні.

2.1 Приховування і відновлення інформаційних повідомлень

Позначимо інформаційне повідомлення як послідовність $m_0, m_1, ..., m_{k-1}$ з окремих бітів, записаних в полярному вигляді:

$$\forall i \in \{0,1,...,k-1\} : m_i \in \{-1,1\}.$$

Для реалізації технології прямого розширення спектра використовуються дискретні сигнали:

$$\Phi_i \in \Phi = \left\{\Phi_0, \Phi_1, ..., \Phi_{M-1}\right\},\ k \le M,$$

причому кожен сигнал являє собою псевдовипадкову послідовність:

$$\forall i \in \{0,1,...,M-1\} : \Phi_i = \left(\varphi_{i_0}, \varphi_{i_1}, ..., \varphi_{i_{n-1}}\right),$$
$$\forall j \in \{0,1,...,n-1\} : \varphi_{i_j} \in \{-1,1\}.$$

База дискретних сигналів B задає кратність розширення спектра:

$$B = TF,$$

де T - тривалість одного елементарного сигналу φ_{i_j}, F - смуга частот сигналу Φ_i.

Для послідовностей з безлічі Φ маємо:

$$F = n\frac{1}{T},$$

Звідки

$$B = n >> 1,$$

тобто використання $\Phi_i \in \Phi$ дозволяє в n раз розширити спектр частот переданих сигналів.

Передбачається, що різні сигнали з безлічі Φ є слабко корельовані, тобто коефіцієнт їх взаємної кореляції приблизно дорівнює нулю:

$$\forall i \ne j : \rho\left(\Phi_i, \Phi_j\right) = \sum_{u=0}^{n-1} \varphi_{i_u} \varphi_{j_u} \approx 0.$$

Стегано-зображення S формується за допомогою додавання до вихідного зображення C посиленого модульованого сигналу E (1):

$$S = C + G \cdot E, \quad (1)$$

де:

$$E = \sum_{i=0}^{k-1} m_i \Phi_i;$$

$G > 0$ - коефіцієнт посилення, який задає «потужність» модульованого сигналу E.

Відновлення інформаційного повідомлення на приймальній стороні здійснюється за допомогою кореляційного прийому. При цьому передбачається, що кожен сигнал з безлічі Φ, є не корельований з вихідним зображенням (2):

$$\forall i : \rho(\Phi_i, C) \approx 0. \quad (2)$$

Тоді значення коефіцієнта кореляції визначається як:

$$\rho(\Phi_i, S) = \rho(\Phi_i, C + G \cdot E) = \rho(\Phi_i, C) + G \cdot \rho(\Phi_i, E) \approx G \cdot \sum_{j=0}^{k-1} m_j \sum_{u=0}^{n} \varphi_{i_u} \varphi_{j_u}.$$

Для всіх $j \neq i$ остання сума:

$$\sum_{u=0}^{n} \varphi_{i_u} \varphi_{j_u} \approx 0,$$

отже, маємо: $\rho(\Phi_i, S) \approx G \cdot m_i \cdot n$, тобто знак $\rho(\Phi_i, S)$ збігається зі значенням m_i (3):

$$m_i = sign\left(\rho(S, \Phi_i)\right) = \begin{cases} -1, \ \rho(S, \Phi_i) < 0; \\ +1, \ \rho(S, \Phi_i) > 0. \end{cases} \quad (3)$$

У роботах [12-15] проведено експериментальні дослідження, в яких показано, що відновлене за формулою (3) повідомлення містить багато помилок. Дійсно, як видно із табл. 2 [14], інтенсивність бітових помилок знаходиться в діапазоні від 10% до 30%. При цьому нижче 10% помилок не вдалося домогтися навіть використовуючи в (1) дуже високий коефіцієнт посилення G. На нашу думку, основною причиною такого високого рівня помилок є невиконання, в більшості випадків, базового припущення (2). Справді, таке припущення може виконуватися в тому випадку, якщо C є реалізацією природного шуму в КЗ. Однак в даному прикладі під C розуміється *реалістичне зображення* в *цифровому вигляді*. Окремі пікселі цього зображення досить сильно корелюють між собою, і це може призвести до ситуації, коли модуль коефіцієнта кореляції буде значно більше нуля:

$$\left|\rho(\Phi_i, C)\right| >> 0.$$

Якщо

$$\left|\rho(\Phi_i, C)\right| > G \cdot n \quad (4)$$

і одночасно

$$sign\left(\rho(\Phi_i, C)\right) \neq m_i,$$

тоді гарантовано відбудеться помилка в цьому інформаційному біті m_i.

У роботах [12-15] для зменшення помилок у відновлених за правилом (3) інформаційних бітах пропонувалося використовувати складні методи корекції помилок, які засновані на внесенні додаткової надмірності. Це знижує швидкість передачі даних, крім того, підвищує складність обробки на приймальній стороні (*або при зчитуванні з носія даних*).

Авторами даної роботи пропонується інший підхід, коли правило формування дискретних сигналів (*послідовностей з безлічі Ф*) враховує статистичні властивості контейнера C. Іншими словами пропонується використання принципу адаптивної генерації, оскільки кожна ПВП з безлічі Φ формується, адаптуючись виключно під локальну статистику даних контейнера C.

2.2 Адаптивна генерація псевдовипадкових послідовностей

Для реалізації адаптивної генерації послідовностей буде введено обмеження на модуль коефіцієнта кореляції контейнера і формованого сигналу:

$$\forall i : \left|\rho(\Phi_i, C)\right| \leq \rho_{\max}. \quad (5)$$

Значення $\rho_{\max}$ визначає максимально допустиму схожість контейнера C на формований сигнал Φ_i (або на його інверсію $-\Phi_i$). Однак на практиці величина $\rho_{\max}$ не може бути занадто малою, тому що час пошуку потрібних псевдовипадкових послідовностей Φ_i може бути дуже великим. Дійсно, кожен сигнал Φ_i з безлічі Φ формується псевдовипадковим чином (використовуючи для цього генератор псевдовипадкових чисел). При чому використовуються не всі формовані послідовності Φ_i, а тільки ті з них, які задовольняють обмеженню (5). Фактично, відбраковуються ті сигнали, для яких $\left|\rho\left(\Phi_i, C\right)\right| > \rho_{\max}$, і при малих значеннях $\rho_{\max}$ частка відбракованих Φ_i різко зростає.

Слід зазначити, що в разі, коли $\rho_{\max} < G \cdot n$ і одночасно $\forall i \neq j : \rho\left(\Phi_i, \Phi_j\right) = 0,$ то буде забезпечено безпомилкове відновлення потайного інформаційного повідомлення.

Дійсно, в цьому випадку всі сигнали з безлічі Φ взаємно ортогональні, тобто виключається виникнення додаткових взаємних перешкод. Умова (4) не може бути виконана і помилки у відновлених за правилом (3) бітах неможливі (*звичайно, ці міркування справедливі тільки за умови відсутності можливих викривлень стегано-зображення* S).

Процес приховування і відновлення інформаційних повідомлень реалізуються так само, як і у відомих способах, тобто з використанням співвідношення (1), (3). Разом з тим тепер використовувані послідовності з безлічі Φ будуть дійсно некорельовані із зображенням C, тобто припущення (2) буде виконуватися (*зазвичай, для малих* $\rho_{\max}$). Нижче показано, що адаптивна генерація сигналів з безлічі Φ дійсно дозволяє істотно знизити інтенсивність помилок у відновленому повідомленні.

3 Експериментальні дослідження

Для підтвердження вірогідності теоретичної частини були проведені експериментальні дослідження. З використанням системи комп'ютерної математики *Mathcad®* (*від фірми MathSoft*) були реалізовані алгоритми вбудовування та вилучення інформаційних повідомлень в контейнер-зображення з використанням виразів (1) і (2).

Для дослідження процесу приховування повідомлень було обрано напівтонове зображення розміром 256×169 елементів з 8-ми бітовим кодуванням кожного з них (Рис. 1). Для приховування *кожного* інформаційного біту використовувався дискретний сигнал - ПВП довжини $n = 256$. Використовуючи правило (1) було послідовно приховано в кожному з 169 рядків контейнера по $k = 1, 2, ..., 256$ інформаційних бітів. Таким чином, в якості C використовувався один з рядків контейнера і цей експеримент повторювався 169 разів. На наведених нижче графіках вказуються усереднені значення інтенсивності помилок.

Так, на рис. 1 наведено зображення, де:

a) вихідне зображення;

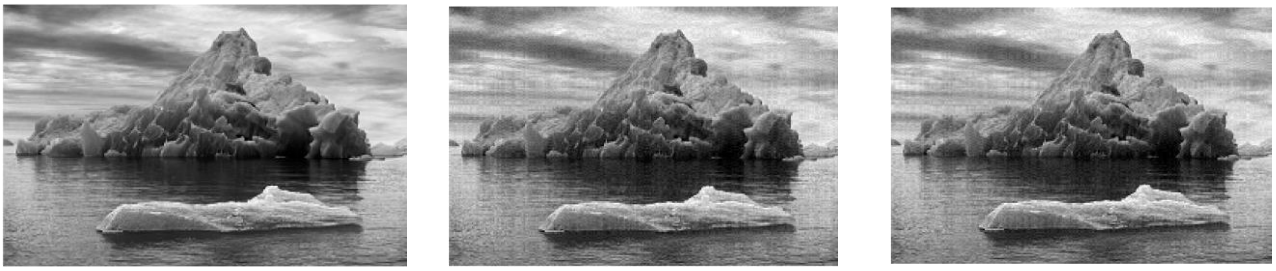
b) зображення після приховування $k = 4$ інформаційних бітів в кожен з рядків контейнера з $G = 4$ *(при цьому використовувалися випадково згенеровані дискретні сигнали)*;

c) зображення після приховування $k = 4$ інформаційних бітів в кожен з рядків контейнера з $G = 4$ *(при цьому використовувалися адаптивно згенеровані дискретні сигнали)*.

Як видно з наведених зображень випадки (b) та (c) *візуально не відрізняються*, тобто адаптивна генерація дискретних сигналів не призводить до підвищення видимих викривлень контейнерів, що використовуються для стегановставки.

В якості дискретних сигналів з множини Φ, використовувалися ПВП, що були зформовані генераторами випадкових чисел *Mathcad®*:

- в разі (b) використовувалися послідовності з рівномірно розподіленими на безлічі {–1,1} послідовностями;
- у разі (c), додатково використовувалось відбраковування дискретних сигналів за правилом (5).

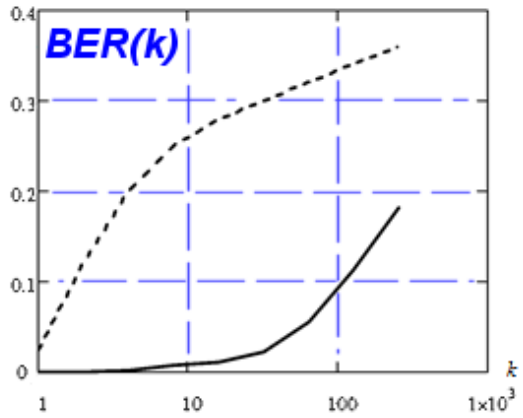


a) *b)* *c)*

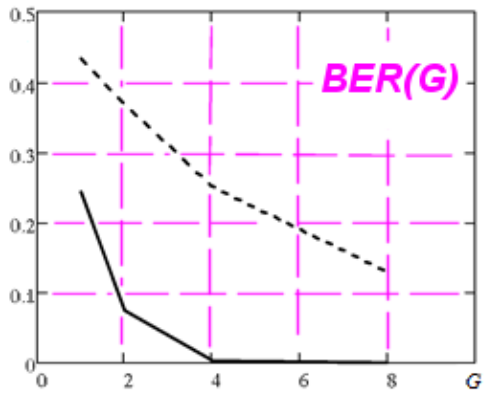
Рис. 1 – Приклади контейнерів-зображень
(помітність спотворень зображення-контейнера)

Найбільш цікавими, на нашу думку, представляються результати досліджень інтенсивності бітових помилок в відновлених повідомленнях. Так, на рис. 2-3 наведені графічні залежності BER для різних умов. Зокрема, на цих рисунках наведені отримані емпіричні залежності BER при відновленні повідомлень за правилом (3), тобто:

- без адаптивної генерації дискретних сигналів (*пунктирна лінія*);
- з використанням адаптивної генерації (*безперервна лінія*).



BER (k)



BER (G)

Рис. 2-3 – BER (k) та BER (G)

При побудові залежностей на рис. 2 використовувалися значення $G = 4$, а k змінювалося в діапазоні від 1 до 255. Як слідує з рис. 2, використання адаптивної генерації, дійсно, дозволяє істотно знизити інтенсивність бітових помилок. Так, навіть при приховуванні 100 і більше інформаційних біт при $G = 4$ вдається досягти низької інтенсивності помилок (*10% і менше*). В контексті цього, нагадаємо, що в роботі [14] було вказано, що таких низьких значень BER не вдавалося досягти навіть при дуже високих значеннях G. Наприклад, при $G = 100$ отримана інтенсивність помилок більше 11% (табл. 2 [14]).

Для залежностей, що відображені на рис. 3, використовувалися значення $k = 4$, а G змінювалося в діапазоні від 1 до 8. При цьому, для розглянутих випадків підвищення коефіцієн-

та посилення дозволяє знизити BER. Однак збільшення G призводить до викривлення зображення-контейнера (див. рис. 1) і т.ч. при $G > 8$ не має практичної доцільності.

Використання адаптивної генерації дискретних сигналів дозволяє істотно знизити BER. Наприклад, при $G > 4$ практично досягається безпомилкове відновлення інформаційних повідомлень. Дійсно, в експериментах були використано $\rho_{\max} = 1000$, що для прийнятих вихідних даних ($G = 4$, $n = 256$) приводить до співвідношення:

$$\rho_{\max} = 1000 < G \cdot n = 1024.$$

Помилки, проте, іноді виникають, тому що застосовані дискретні сигнали не ортогональні (*вони квазіортогональні*):

$$\forall i \neq j : \rho\left(\Phi_i, \Phi_j\right) \approx 0,$$

але інтенсивність помилок істотно знижена (*див. Рис. 2-3*).

При подальшому збільшенні коефіцієнта посилення (тобто при $G > 4$), помилки практично виключені (*див. Рис. 3*).

4 Висновок

В даній роботі було досліджено технологію прямого розширення спектра та її застосування в інтересах стеганографії. Зокрема, розглянуті способи приховування інформаційних повідомлень в зображенні-контейнері, з використанням довгих ПВП (*складних дискретних сигналів з базою* $B = n >> 1$). Основне припущення для побудови таких стеганосистем полягає у відсутності кореляції зображень і використовуваних дискретних сигналів.

Справді, традиційно зображення інтерпретується, як природний шум в КЗ, та для такої інтерпретації, очевидною виглядає статистична незалежність дискретних сигналів і контейнерів. Однак, висока інтенсивність помилок у відновлених повідомленнях спростовує це припущення і наше моделювання це наочно підтверджує.

Пропонується новий підхід, що полягає в адаптивному формуванні дискретних сигналів. В цьому разі, якщо правило формування ПВП буде враховувати статистичні властивості контейнеру, то тоді можна забезпечити виконання базового припущення *про відсутність кореляції зображень і дискретних сигналів*. В межах роботи стверджується, що таку генерацію можна реалізувати, наприклад, найпростішим відбракуванням послідовностей. Для цього потрібно лише ввести обмеження на допустиму корельованість сигналів і зображень, та використовувати, в подальшому, тільки відповідні послідовності.

Проведені експериментальні дослідження підтвердили відповідні теоретичні твердження. Вдалося істотно знизити інтенсивність помилок у відновлених повідомленнях. При цьому, викривлення контейнера залишилися на колишньому рівні.

Важливо зазначити, що використання адаптивної генерації дозволяє реалізувати і безпомилкове відновлення повідомлень. Для цього необхідно вибрати досить суворі обмеження, як на корельованість сигналів і контейнерів, так і на взаємну подобу сигналів один з одним.

Дані експерименти підтверджують практично повну відсутність помилок, що наочно підтверджує достовірність наведених суджень.

Перспективним напрямком подальших досліджень слід розглядати використання дискретних сигналів з особливими кореляційними властивостями, наприклад таких як в [22-26].

Посилання

[1] “Digital Watermarking and Steganography,” 2008. doi:10.1016/b978-0-12-372585-1.x5001-3.

[2] F. Y. Shin, “Digital Watermarking and Steganography,” Dec. 2017. doi:10.1201/9781315219783.

[3] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," in Computer, vol. 31, no. 2, pp. 26-34, Feb. 1998. doi: 10.1109/MC.1998.4655281.

[4] I. V. S. Manoj, “Cryptography and Steganography,” International Journal of Computer Applications, vol. 1, no. 12, pp. 63–68, Feb. 2010. doi:10.5120/257-414.

[5] A. Z. Tirkel, C. F. Osborne and R. G. Van Schyndel, "Image watermarking-a spread spectrum application," Proceedings of ISSSTA'95 International Symposium on Spread Spectrum Techniques and Applications, Mainz, Germany, 1996, pp. 785-789 vol.2. doi: 10.1109/ISSSTA.1996.563231.

[6] J. R. Smith and B. O. Comiskey, “Modulation and information hiding in images,” Lecture Notes in Computer Science, pp. 207–226, 1996. doi:10.1007/3-540-61996-8_42.
[7] M. Kutter, “Performance Improvement of Spread Spectrum Based Image Watermarking Schemes through M-ary Modulation,” Lecture Notes in Computer Science, pp. 237–252, 2000. doi:10.1007/10719724_17.
[8] V. P. Ipatov, “Spread Spectrum and CDMA,” Mar. 2005. doi:10.1002/0470091800.
[9] “Introduction to CDMA Wireless Communications,” 2007. doi:10.1016/b978-0-7506-5252-0.x5001-7.
[10] “The Generalized CDMA,” CDMA: Access and Switching, pp. 1–28. doi:10.1002/0470841699.
[11] S. Hara and R. Prasad, "DS-CDMA, MC-CDMA and MT-CDMA for mobile multi-media communications," Proceedings of Vehicular Technology Conference - VTC, Atlanta, GA, USA, 1996, pp. 1106-1110 vol.2. doi: 10.1109/VETEC.1996.501483.
[12] L. M. Marvel, C. G. Boncelet, R. Jr., and Charles T., “Methodology of Spread-Spectrum Image Steganography,” Jun. 1998. doi:10.21236/ada349102.
[13] L. M. Marvel, C. G. Boncelet and C. T. Retter, "Spread spectrum image steganography," in IEEE Transactions on Image Processing, vol. 8, no. 8, pp. 1075-1083, Aug. 1999. doi: 10.1109/83.777088.
[14] F. S. Brundick and L. M. Marvel, “Implementation of Spread Spectrum Image Steganography,” Mar. 2001. doi:10.21236/ada392155.
[15] Patent No.: US 6,557,103 B1, Int.Cl. G06F 11/30. Charles G. Boncelet, Jr., Lisa M. Marvel, Charles T. Retter. Spread Spectrum Image Steganography. Patent No.: US 6,557,103 B1, Int.Cl. G06F 11/30. – № 09/257,136; Filed Feb. 11, 1999; Date of Patent Apr. 29, 2003
[16] Fan Zhang, Bin Xu and Xinhong Zhang, "Digital image watermarking algorithm based on CDMA spread spectrum," 2006 12th International Multi-Media Modelling Conference, Beijing, 2006, pp. 4 pp.-. doi: 10.1109/MMMC.2006.1651359.
[17] T. T. Nguyen and D. Taubman, "Optimal linear detector for spread spectrum based multidimensional signal watermarking," 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, 2009, pp. 113-116. doi: 10.1109/ICIP.2009.5414121.
[18] E. Nezhadarya, Z. J. Wang and R. K. Ward, "Image quality monitoring using spread spectrum watermarking," 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, 2009, pp. 2233-2236. doi: 10.1109/ICIP.2009.5413955.
[19] S. Ghosh, P. Ray, S. P. Maity and H. Rahaman, "Spread Spectrum Image Watermarking with Digital Design," 2009 IEEE International Advance Computing Conference, Patiala, 2009, pp. 868-873. doi: 10.1109/IADCC.2009.4809129.
[20] H. O. Altun, A. Orsdemir, G. Sharma and M. F. Bocko, "Optimal Spread Spectrum Watermark Embedding via a Multistep Feasibility Formulation," in IEEE Transactions on Image Processing, vol. 18, no. 2, pp. 371-387, Feb. 2009. doi: 10.1109/TIP.2008.2008222.
[21] A. Samčović and M. Milovanović, "Robust digital image watermarking based on wavelet transform and spread spectrum techniques," 2015 23rd Telecommunications Forum Telfor (TELFOR), Belgrade, 2015, pp. 811-814. doi: 10.1109/TELFOR.2015.7377589.
[22] Yu.V. Stasev, A.A. Kuznetsov, A.M. Nosik. “Formation of pseudorandom sequences with improved autocorrelation properties.” Cybernetics and Systems Analysis, vol. 43, Issue 1, pp. 1-11, January 2007. DOI: 10.1007/s10559-007-0021-2
[23] N.I.Naumenko, Yu.V.Stasev, A.A.Kuznetsov. “Methods of synthesis of signals with prescribed properties.” Cybernetics and Systems Analysis, vol. 43, Issue 3, pp. 321-326, May 2007. DOI: 10.1007/s10559-007-0052-8
[24] O.Karpenko, A.Kuznetsov, V.Sai, Yu.Stasev. “Discrete Signals with Multi-Level Correlation Function.” Telecommunications and Radio Engineering, vol. 71, 2012 Issue 1. pp 91-98. DOI: 10.1615/TelecomRadEng.v71.i1.100
[25] A. Kuznetsov, S. Kavun, V. Panchenko, D. Prokopovych-Tkachenko, F. Kurinniy and V. Shoiko, "Periodic Properties of Cryptographically Strong Pseudorandom Sequences," 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, Ukraine, 2018, pp. 129-134. doi: 10.1109/INFOCOMMST.2018.8632021
[26] A. Kuznetsov, O. Smirnov, D. Kovalchuk, A. Averchev, M. Pastukhov and K. Kuznetsova, "Formation of Pseudorandom Sequences with Special Correlation Properties," 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), Lviv, Ukraine, 2019, pp. 395-399. doi: 10.1109/AIACT.2019.8847861

Reviewer: Vyacheslav Kalashnikov, Dr. of Sciences (Physics and Mathematics), Full Prof., Department of Systems and Industrial Engineering, Campus Monterrey, Monterrey, Mexico.
E-mail: kalash@itesm.mx

Received on January 2020.

Authors:
Eugene Demenko, Computer Science Student, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine,
E-mail: demenjay@gmail.com

Alexander Onikiychuk, Computer Science Student, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: onik4524a@gmail.com

Anna Arischenko, Computer Science Student, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine, Ukraine.
E-mail: annaarischenko@gmail.com

Ludmila Gorbachova, Computer Science Student, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine, Ukraine.
E-mail: lusyag23@gmail.com

Oleksii Smirnov, Central Ukrainian National Technical University, Cybersecurity & Software Academic Department, Kropivnitskiy, Ukraine. E-mail: dr.smirnovoa@gmail.com

Adaptive Pseudo-Random Sequence Generation for Spread Spectrum Image Steganography.

Abstract. In this article we consider the ways of data hiding in digital images with the use of pseudorandom sequences and the spread spectrum technique. We propose a new way of the generation of sequences, which considers statistical properties of cover-images. This makes it possible to achieve a low correlation, which provides reliable and safe data hiding in digital images. The results of experimental researches show, that the bit error rate in restored messages is significantly reduced. At the same time, the distortions of cover-images remain the same.

Keywords: Steganography, Data hiding, Digital images, Pseudorandom sequences, Spread spectrum image steganography.

Рецензент: Вячеслав Калашников, д.ф.-м.н., проф., Технологический университет Монтеррея, Монтеррей, Мексика.
E-mail: kuznetsov@karazin.ua

Поступила: Январь 2020.

Авторы:

Евгений Деменко, студент факультета компьютерных наук, ХНУ имени В. Н. Каразина, Харьков, Украина.
E-mail: demenjay@gmail.com

Анна Арищенко, студентка факультета компьютерных наук, ХНУ имени В. Н. Каразина, Харьков, Украина.
E-mail: annaarischenko@gmail.com

Александр Оникийчук, студент факультета компьютерных наук, ХНУ имени В. Н. Каразина, Харьков, Украина.
E-mail: onik4524a@gmail.com

Людмила Горбачова, студентка факультета компьютерных наук, ХНУ имени В. Н. Каразина, Харьков, Украина.
E-mail: lusyag23@gmail.com

Алексей Смирнов, кафедра кибербезопасности и программного обеспечения, Центральный украинский национальный технический университет, Кропивницкий, Украина.
E-mail: dr.smirnovoa@gmail.com

Формирование псевдослучайных последовательностей для сокрытия данных в изображениях.

Аннотация. В статье рассматриваются способы сокрытия данных в цифровых изображениях, с использованием псевдослучайных последовательностей и технологии прямого расширения спектра. Предлагается новый способ формирования последовательностей, который учитывает статистические свойства изображений-контейнеров. Это позволяет добиться низкой корреляции, обеспечивает надежное и безопасное сокрытие информации в цифровых изображениях. Результаты экспериментальных исследований свидетельствуют, что интенсивность битовых ошибок в восстановленных сообщениях, существенно снижена. При этом искажения контейнеров изображений, остаются на прежнем уровне.

Ключевые слова: стеганография; сокрытие данных; цифровые изображения; псевдослучайная последовательность; технология прямого расширения спектра.

УДК 621.327:621.391 DOI: 10.26565/2519-2310-2019-4-05

ОПИСАНИЕ СХЕМЫ API ТЕСТИРОВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Ольга Мелкозерова, Валерия Гайкова, Сергей Малахов

Харьковский национальный университет имени В.Н. Каразина, пл. Свободы, 4, Харьков, 61022, Украина
olja.mex@gmail.com, valeriagaikova98@gmail.com, mailgate@meta.ua

Рецензент: Георгий Кучук, д.т.н., проф., НТУ «ХПИ», ул. Кирпичова, 21, Харьков, 61000, Украина.
kuchuk56@ukr.net

Поступила: Январь 2020

Аннотация. *Для оценки качества приложений применяют API тестирование. Существует большое количество инструментов для его проведения (Postman, SoapUI, Jmeter и т.п.). Универсальный подход к тестированию усложняется большими объемами данных, наличием разнообразных методик и инструментов, также необходим объект тестирования. Сложность логики работающего приложения приводит к трудностям понимания процесса тестирования, то есть это затрагивает вопрос подготовки специалистов в этой области. Для упрощения понимания, в качестве объекта тестирования было создано приложение «Калькулятор» при помощи класса Servlet (Java). Тестирование проводилось при помощи инструмента Jmeter и написания кода на языке Java. Составлен тест-план с использованием инструмента JMeter, позволяющий отправлять запрос приложению. Отмечено, что автоматизированные модульные тесты находят большее применение при разработке программного обеспечения, поэтому на языке программирования Java показана возможность написания аналогичного запроса.*

Ключевые слова: *автоматизированное тестирование; инструменты автоматизированного тестирования; технологии автоматизированного тестирования; JMeter.*

1 Введение

Многие современные приложения используют API (*Application Programming Interface*) для обеспечения их взаимодействия с сервером и интеграции друг с другом, поэтому для их тестирования необходимо владеть инструментами и техниками тестирования API, которое обладает рядом преимуществ [1]:

1. *Раннее тестирование* – разработчики приложений сначала делают API, а потом уже графический интерфейс GUI (*Graphical User Interface*). При этом обеспечивается возможность проверить логику раньше, чем появится GUI.
2. Графического интерфейса может в принципе не быть.
3. Высокая скорость – вызов одного запроса занимает доли секунды. При этом через интерфейс повторить процедуру бывает сложно.
4. Возможность автоматизации – даже если нет авто тестов на уровне API приложения, их всегда можно создать вручную, посредством PostMan и JMeter [2, 3].
5. GUI и его содержание постоянно меняется, однако логика работы приложения остается прежней.

В данной работе на примере простейшего приложения «Калькулятор» демонстрируется возможность тестирования API путем написания автоматизированных тестов на языке программирования Java при помощи инструмента JMeter.

2 Сборка приложения «Калькулятор» при помощи Servlet

Для демонстрации возможностей тестирования производительности и тестирования API на языке Java при помощи Servlet (*сервлет*) было собрано простейшее приложение «Калькулятор». Сервлет – это класс, который расширяет функциональность класса HttpServlet и запускается внутри контейнера сервлетов [4].

Сервлет размещается на сервере, однако чтобы сервер мог использовать сервлет для обработки запросов, сервер должен поддерживать движок или контейнер сервлетов (*Servlet Container/Engine*). Например, Apache Tomcat по сути является контейнером сервлетов,

поэтому он может использовать сервлеты для обслуживания запросов.

Для обработки запроса в HttpServlet определен ряд методов, которые мы можем переопределить в классе сервлета:

- *doGet*: обрабатывает запросы GET (*получение данных*);
- *doPost*: обрабатывает запросы POST (*отправка данных*);
- *doPut*: PUT (отправка данных для изменения);
- *doDelete*: DELETE (*удаление данных*);
- *doHead*: обрабатывает запросы HEAD.

Для создания приложения используем сборщик проектов Maven. Maven - фреймворк для автоматизации и сборки проектов на основе описания их структуры в файлах на языке POM (*Project Object Model*), который является подмножеством XML [5].

Структура проекта отображена ниже, на рис. 1, а скриншот приложения, которое подлежит тестированию, представлен на рис. 2.

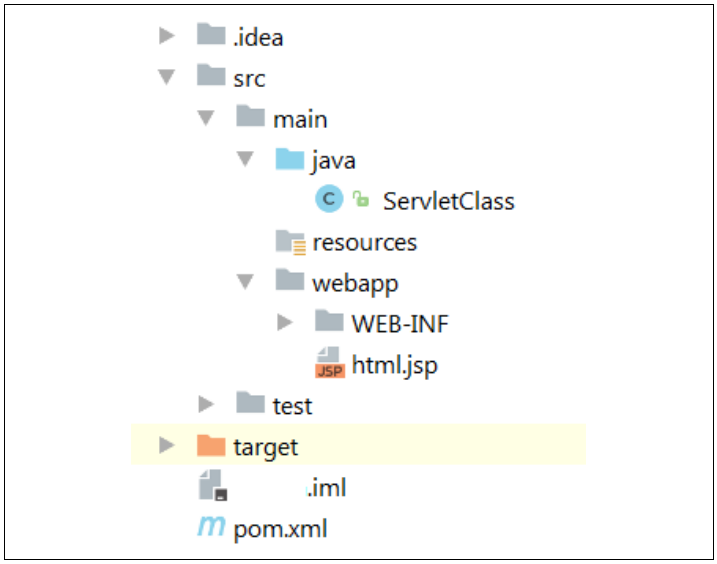


Рис. 1 – Структура проекта *Servlet*

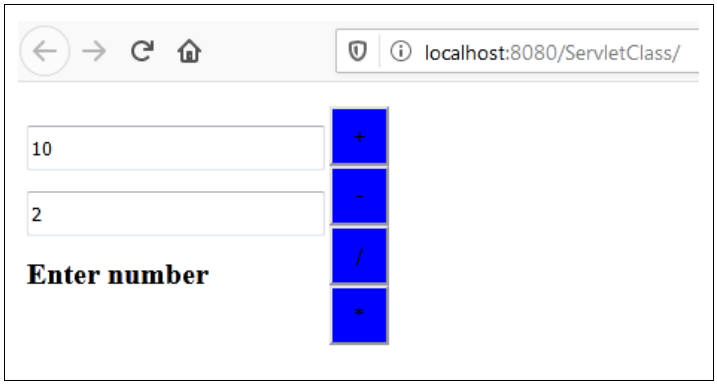


Рис. 2 – Скриншот работающего приложения «*Калькулятор*»

3. Написание API запросов к приложению

3.1 Использование приложения Jmeter для тестирования производительности и написания запросов.

Apache JMeter – открытое программного обеспечение, которое используется Java для тестирования функционального поведения и оценки производительности приложений [2].

Рассмотрим пример, в котором для исследования производительности приложения использован Apache JMeter версии 2.11.20151206 (см. Рис. 3).

Как было отмечено выше, тестированию подвергалось приложение «Калькулятор», тест-план которого представлен на Рис. 3-4. Перечень элементов, из которых возможен синтез тест-планов в JMeter выглядит следующим образом:

- Группы потоков (*Thread groups*);
- Логические контролеры (*Logic controller*);
- Типовые контроллеры (*Sample generating controller*);
- Слушатели (*Listeners*);
- Таймеры (*Timers*);
- Соответствия (*Assertions*);
- Конфигурационные элементы (*Configuration elements*).

Thread groups – начальные точки любого тест-плана (Рис. 3). Все контроллеры и образцы должны быть в группе потоков. Другие элементы, такие как слушатели, могут располагаться под тест-планом, в котором они применяются для всех потоков групп. Элемент группы потоков управляет количеством потоков, который JMeter будет использовать для выполнения теста.

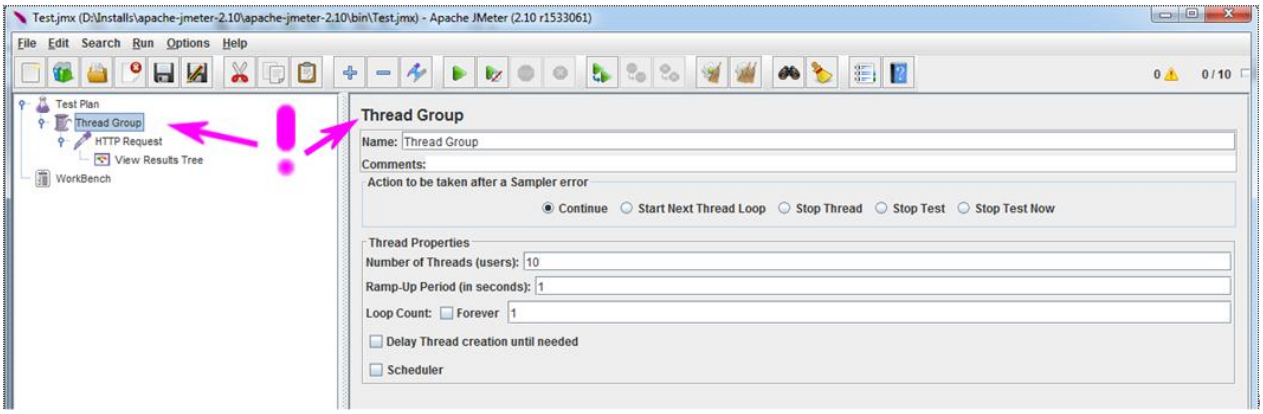


Рис. 3 – Заполнение формы Thread Group *(количество пользователей – 10)*

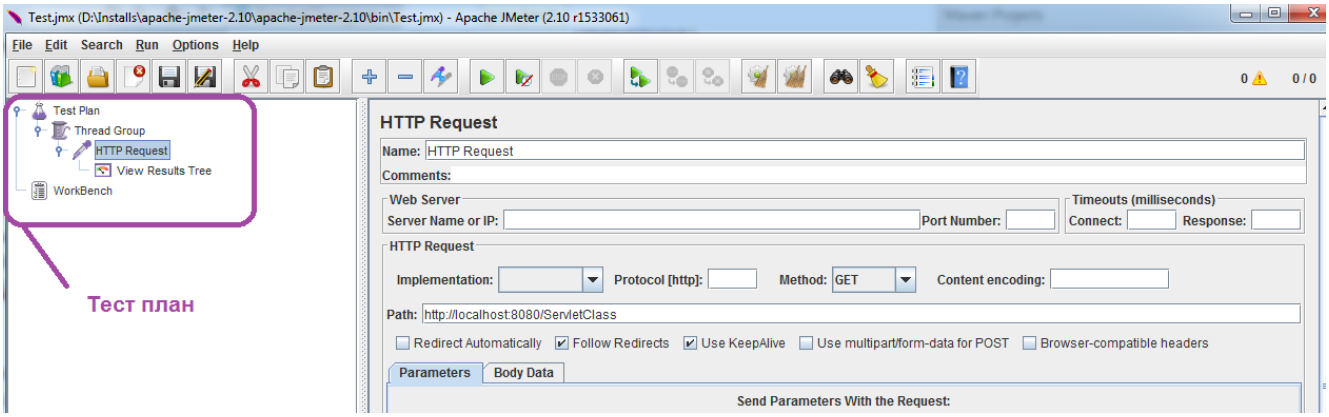


Рис. 4 – Заполнение формы HTTP запроса

Для отправки запроса необходимо заполнить форму HTTP запроса (Рис. 4), в котором следует указать ссылку на работающее приложение *(http://localhost:8080/ServletClass/).*

Listener – это компонент, который показывает результаты образцов. Сами результаты могут отображаться в дереве, таблице, графах или быть записаны в лог-файл. В данной статье для записи результата использовался Listener *Tree Graph Results.*

Tree Graph Results – дерево ответов всех образцов, позволяет увидеть отклик для всех образцов. В дополнение к показу отклика, индицируется время этого отклика (см. *вкладку «Sampler Result» на Рис. 5*), код отклика («200»), *HTML формат приложения, Рис. 6.*

Jmeter используется для написания API запросов, но его также используют для тестирования производительности. В целом могут быть определены различные типы тестов, которые зависят от целей тестирования [1, 6].

Термин «*тестирование производительности*» (*Performance Testing*) включает в себя любое тестирование, которое сфокусировано на производительности системы (*ответной реакции, скоростных показателях*) или компонентов под различными объемами и характерами нагрузки.

Нагрузочное тестирование (*Load Testing*) фокусируется на способности системы справляться с возрастающими уровнями ожидаемой нагрузки, возникающей в результате запросов конкурирующих пользователей или процессов. Кроме того, это и исследование робастности (*запаса прочности*) – способности системы сохранять заданные показатели качества, как при допустимых пределах нагрузки, так и при их некотором превышении.

Стрессовое тестирование (*Stress Testing*) сосредоточено на способности системы или компонентов поддерживать пиковые нагрузки, которые находятся на уровне или превышают

пределы ожидаемых. Фактически, это исследование поведения приложения при «аномальных» изменениях нагрузки. Данный тип тестирования также используется для того, чтобы оценить способность системы поддерживать такие показатели, как: - доступная компьютерная мощность; - доступная полоса пропускания; - доступная память.

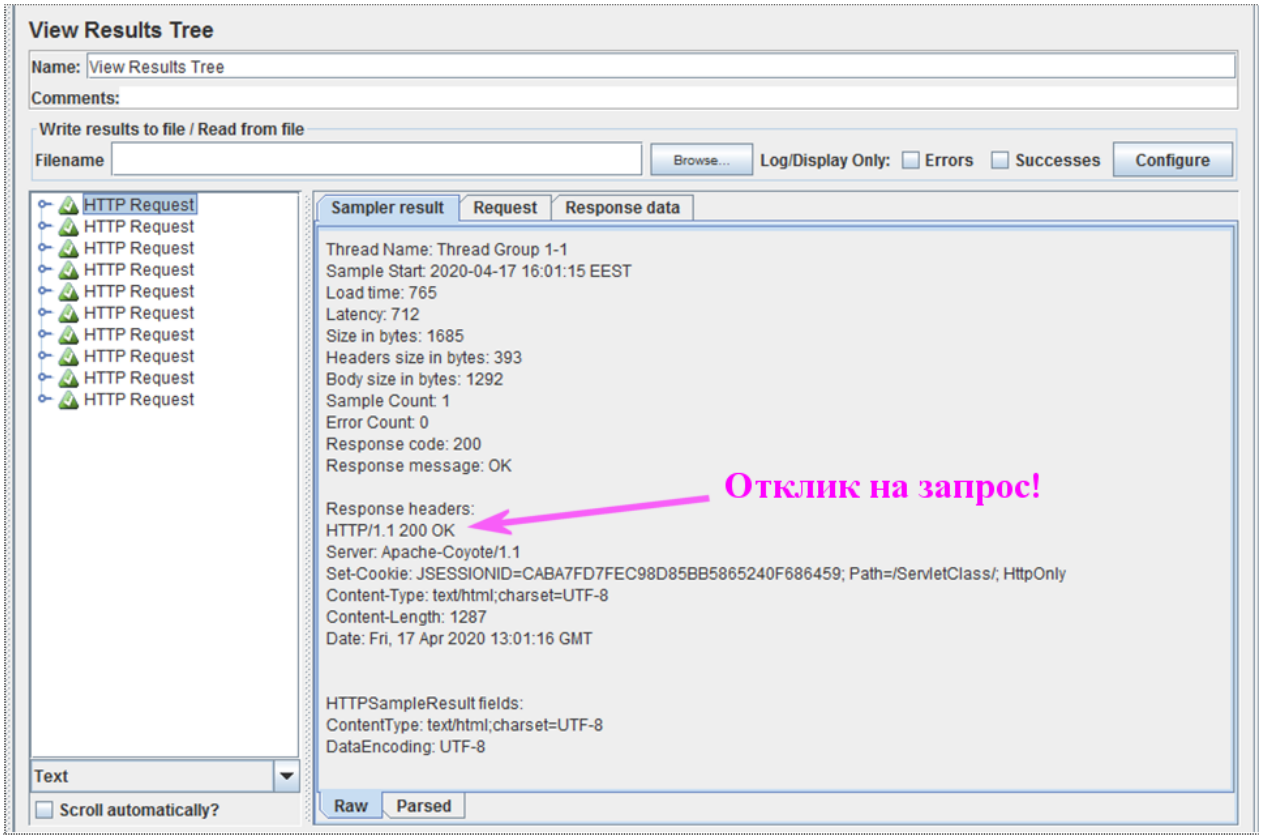


Рис. 5 – Результат GET запроса

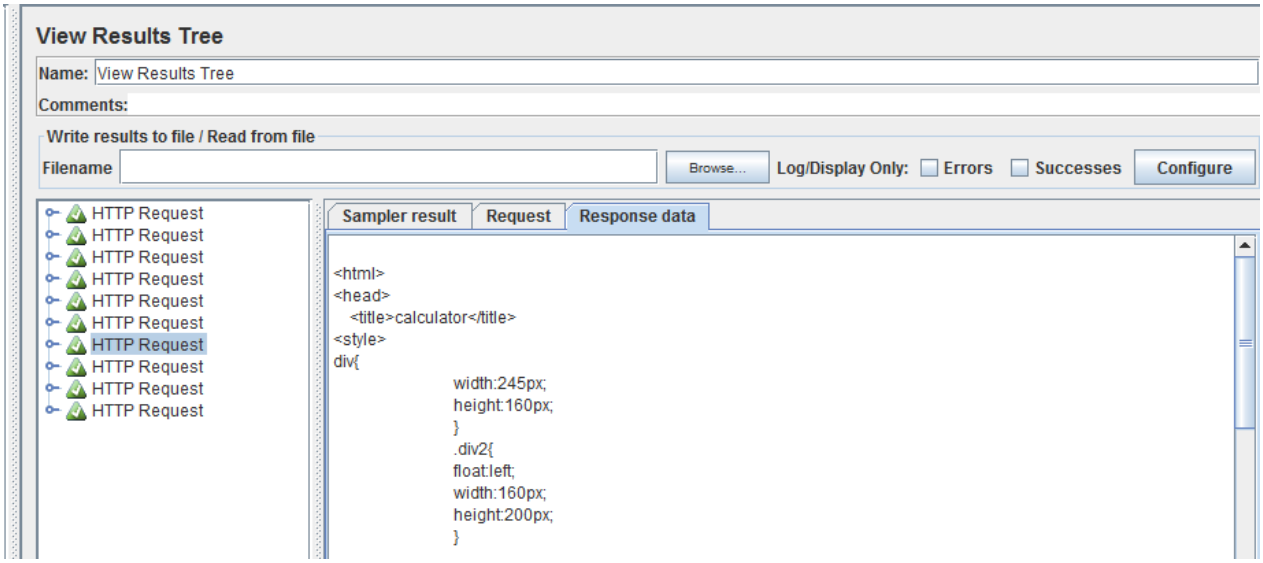


Рис. 6 – Результат GET запроса *(виден HTML-текст приложения)*

Тестирование масштабируемости (*Scalability Testing*) проверяет способность системы увеличивать свою производительность. Цель этих тестов – определить способность системы «расти» (*например, с большим количеством пользователей или с большим объемом данных*) без нарушения заданных для нее требований по производительности. Другими словами, это

исследование способности приложения *увеличивать показатели производительности в соответствии с увеличением количества доступных приложению ресурсов*.

Объемное тестирование (*Volume Testing*) – исследование производительности системы при обработке различных объемов данных.

Всплесковое тестирование (*Spike Testing*) – основывается на способности системы корректно отвечать при внезапных всплесках пиковых нагрузок и ее последующему возвращению к номинальному состоянию.

Тестирование на выносливость (*Endurance Testing*) фокусируется на стабильности системы. Этот тип тестирования проверяет наличие проблем с ресурсами (*например, нехватка памяти, соединение с базой данных, потоками*), которые влияют на производительность, или могут служить причинами сбоев в критических точках.

Тестирование надежности (*Reliability Testing*) – очень близкое по своему смыслу к предыдущему виду тестирования. Это проверка способности приложения выполнять свои функции в заданных условиях на протяжении заданного времени или заданного количества операций.

Конкурентное тестирование (*Concurrency Testing*) фокусируется на влиянии ситуации, при которой одновременно происходят множественные специфические действия (*например, одновременная работа большого количества пользователей*). При этом тестировании исследуется поведение системы в ходе обработке большого количества одновременно поступающих запросов, что вызывает конкуренцию между запросами за имеющиеся ресурсы (*базу данных, память, канал передачи данных, дисковую систему*). Иногда под конкурентным тестированием, также, понимают исследование работы многопоточного приложения и корректность синхронизации действий.

Тестирование мощности (*Capacity Testing*) определяет, как много пользователей может обслуживать система.

В целом при тестировании производительности следует различать статическое и динамическое тестирование. Статическое тестирование, зачастую, более важно для тестирования производительности, чем тестирование функциональной пригодности. Это является следствием того, что очень много критических дефектов вносятся в архитектуру и дизайн системы еще на этапе ее проектирования. Дефекты могут возникать из-за недопонимания или недостатка знаний дизайнеров и архитекторов. Кроме того, функциональные требования, цели использования, ожидаемая нагрузка и существующие ограничения могут быть недостаточно изучены по причине банального дефицита времени.

После запуска тест-плана, можно увидеть соответствующий отклик на запрос *(сообщение «200 OK» на Рис. 5)*.

3.2 Формирование API запроса на языке программирования Java

С целью формирования GET запроса на языке программирования Java, для работающего приложения, достаточно написать следующий код (*структура проекта приведена на рис. 7*):

```
public class Http {
    private static final Logger LOG = LogManager.getLogger(Http.class);
    public static void main(String[] args) throws IOException, SAXException,
ParserConfigurationException {
        String query = "http://localhost:8080/ServletClass";
        HttpURLConnection connection = null;
        try {
            connection = (HttpURLConnection) new URL(query).openConnection();
            connection.setRequestMethod("GET");
            connection.setUseCaches(false);
            connection.setConnectTimeout(2500);
            connection.setReadTimeout(2500);
            connection.connect();
            LOG.info(connection.getResponseMessage());
```

```
            LOG.info(connection.getResponseCode());
            StringBuilder sb = new StringBuilder();
            if (HttpURLConnection.HTTP_OK == connection.getResponseCode()) {
                BufferedReader in = new BufferedReader(new
InputStreamReader(connection.getInputStream()));
                String line;
                while ((line = in.readLine()) != null) {
                    sb.append(line);
                    sb.append("\n");
                }
                String s = sb.toString();
                LOG.info(s);
            }
        } catch (ProtocolException e) {
            e.printStackTrace();
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
```

В результате запуска программы в консоль выведется отклик «OK», код сообщения «200», а также HTML формат интерфейса калькулятора (*см. Рис. 8*).

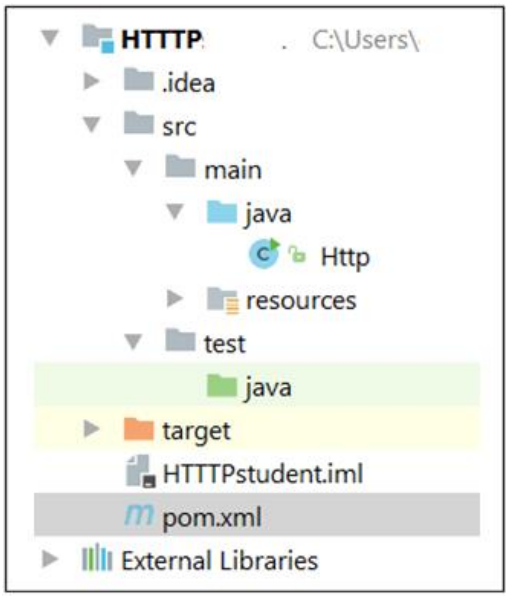


Рис. 7 – Структура проекта

```
"C:\Program Files\Java\jdk1.8.0_131\bin\java" ...
2020-04-19 12:51:21 [INFO ] Http:25 - OK
2020-04-19 12:51:21 [INFO ] Http:26 - 200
2020-04-19 12:51:21 [INFO ] Http:36 -
<html>
<head>
    <title>calculator</title>
<style>
div{
    width:245px;
    height:160px;
    }
    .div2{
    float:left;
    width:160px;
    height:200px;
    }
```

Рис. 8 – Результат написания GET запроса

4 Выводы

1. API тестирование обладает рядом преимуществ по сравнению с GUI тестированием. В работе рассмотрена возможность проведения данного способа тестирования с использованием двух различных подходов: – посредством применением Jmeter; – путем написания кода на языке программирования Java. Отмечено, что JMeter, также, находит свое применение для целей тестирования производительности приложений.

2. Акцентировано внимание на то, что тестирование производительности – сложный и ответственный этап разработки программного обеспечения.

3. Рассмотренная классификация видов тестовых испытаний является, фактически, полнофункциональной замкнутой системой для всесторонней оценки качества программных продуктов.

4. При динамическом системном тестировании приложений, тесты разрабатываются на основе функциональных требований исходя из уровня предполагаемой нагрузки на проектируемую систему.

5. Одним из наиболее эффективных инструментов, обеспечивающим возможность автоматизированной реализации количественной оценки показателей качества программного обеспечения, является приложение JMeter. Основные этапы использования данного инструмента можно рассматривать, как «дорожную карту» для соответствующих специалистов-тестировщиков.

6. Приведенная схема тестирования производительности веб-приложений, иллюстрирует возможность получения количественных оценок для последующего анализа производительности в условиях различных уровней нагрузки.

7. Удобство и практическая ценность JMeter заключается в высокой информативности и наглядности результатов проведенных испытаний (*таблицы и графики*), что создает хорошие условия для всестороннего анализа выполнения требований по производительности.

8. Следует обратить внимание на то, что этап составления тест-плана является особо ответственной процедурой.

9. Существенным преимуществом инструмента JMeter является его приспособленность для целей осуществления API тестирования. Подобное тестирование позволяет не только оценить комфортность перспективной эксплуатации будущего программного продукта, но и выявить возможные дефекты в логике выполнения его вычислительных алгоритмов.

10. Следует обратить внимание на тот факт, что аналогичные тесты можно проводить и с привлечением платформы Intellij Idea Java для написания модульных API тестов, что, на данный момент времени, является более востребованным направлением в тестировании.

Ссылки

[1] Тестирование программного обеспечения. Базовый курс/ Святослав Куликов. [Электронный ресурс] – Режим доступа: http://svyatoslav.biz/software_testing_book/ - 28.12.2017.
[2] Jmeter URL: https://jmeter.apache.org/ - 20.04.2020.
[3] Postman URL: https://www.postman.com/ - 20.04.2020.
[4] URL: https://metanit.com/java/javaee/4.1.php - 19.04.2020.
[5] Maven repository https://mvnrepository.com/ - 20.04.2020.
[6] URL: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010 - 20.04.2020.

Рецензент: Георгій Кучук, д.т.н., проф., НТУ «ХПІ», Харків, Україна.
E-mail: kuchuk56@ukr.net

Надійшло: Січень 2020.

Автори:

Ольга Мелкозьорова, к.т.н., старший викладач кафедри, ХНУ імені В.Н. Каразіна, Харків, Україна.
E-mail: olja.mex@gmail.com

Валерія Гайкова, студентка факультету комп'ютерних наук, ХНУ імені В.Н. Каразіна, Харків, Україна.
E-mail: valeriagaikova98@gmail.com

Сергій Малахов, к.т.н., ст. наук. співробітник, доцент кафедри, ХНУ імені В.Н. Каразіна, Харків, Україна.
E-mail: mailgate@meta.ua

Опис схеми API тестування програмного забезпечення.

Анотація. Для оцінки якості додатків використовують API тестування. Існує велика кількість інструментів для його проведення (Postman, SoapUI, Jmeter і т.і.). Універсальний підхід до тестування ускладнюється великим обсягами даних, наявністю різноманітних методик та інструментів, також повинен бути об'єкт тестування. Складність логіки додатку, що працює, призводить до труднощів розуміння процесу тестування, тобто це стосується питання підготовки фахівців в цій галузі. Для спрощення розуміння в якості об'єкту для тестування було створено додаток «Калькулятор» за допомогою класу Servlet (Java). Тестування проводилося за допомогою інструменту JMeter та написання коду на мові Java. Складено тест-план з використанням інструменту JMeter, що дозволяє відправляти запити додатку. Відзначено, що автоматизовані модульні тести знаходять більше застосування при розробці програмного забезпечення, тому на мові програмування Java показана можливість написання аналогічного запиту.

Ключові слова: автоматизоване тестування; інструменти автоматизованого тестування; технології автоматизованого тестування, JMeter.

Reviewer: George Kuchuk, Doctor of Sciences (Eng.), Full Prof., NTU "KhPI", Kharkiv, Ukraine.

E-mail: kuchuk56@ukr.net

Received on January 2020.

Authors:
Olga Melkozerova, Ph.D., Senior Lecturer of the Department, V. N. Karazin Kharkiv National University, Ukraine.
E-mail: olja.mex@gmail.com
Valeria Gaykova, Computer Science Student, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine, Ukraine.
E-mail: valeriagaikova98@gmail.com
Serhii Malakhov, Ph.D., Senior Research, Associate Prof. of the Department, V. N. Karazin Kharkiv National University, Ukraine.
E-mail: mailgate@meta.ua

Circuit description of API interface for software testing.

Annotation. They use the testing API to evaluate the quality of applications. There are a large number of tools for its implementation (Postman, SoapUI, JMeter, etc.). The universal approach to testing is becomes complicated by the large amount of data, the availability of various techniques and tools, and there should also be a test object. The complexity of the application logic makes it difficult to understand the testing process; this concerns the training of specialists in this field. To simplify understanding, in the article, the Servlet (Java) Calculator application was used as a test object. Testing was done using the Jmeter tool and code writing in Java. A test plan was developed using a JMeter tool that allows you to submit application requests. It is noted that automated unit tests are more useful in software development, so the Java programming language shows the ability to write query algorithms.

Key words: Automated testing; Automated testing tools; Automated testing technologies; JMeter.

УДК 004.056.55 DOI: 10.26565/2519-2310-2019-4-06

НЕДВОИЧНЫЕ КРИПТОГРАФИЧЕСКИЕ ФУНКЦИИ ДЛЯ ГЕНЕРАЦИИ БЛОКОВ ПОДСТАНОВОК СИММЕТРИЧНЫХ ШИФРОВ

Никита Гончаров, Татьяна Кузнецова, Александр Кузнецов

Харьковский национальный университет имени В.Н. Каразина, пл. Свободы, 4, Харьков, 61022, Украина
worldxdark@gmail.com, kuznetsova.tatiana17@gmail.com, kuznetsov@karazin.ua

Рецензент: Сергей Толюпа, д.т.н., проф., Киевский национальный университет имени Т. Шевченко, ул. М. Ломоносова 81, г. Киев, 03189, Украина.
tolupa@i.ua

Поступила: Январь 2020.

Аннотация: *Важным элементом современных симметричных криптоалгоритмов являются нелинейные узлы замен (блоки подстановок, S-блоки). Свойства этих блоков непосредственно влияют на показатели криптографической стойкости алгоритмов шифрования. Например, стойкость симметричных шифров к дифференциальному и линейному криптоанализу непосредственно зависит от показателей нелинейности S-блоков и их корреляционных свойств. В данной статье исследуются способы математического описания регулярных нелинейных узлов замен, вычислительные техники их генерации и оценки криптографических показателей. Рассматривается традиционный подход, в котором для описания внутренней структуры S-блоков используется совокупность компонентных булевых функций. Исследуются недвоичные функции, с помощью которых формируются регулярные нелинейные узлы замены. Приводятся результаты вычислительного поиска S-блоков с использованием предлагаемого подхода. Для генерации регулярных S-блоков использовались методы имитации отжига, применительно к недвоичным функциям с улучшенными ценовыми функциями (настраиваемый элемент метода имитации отжига). Показано, что по нелинейности и автокорреляции, сформированные узлы замен обладают улучшенными свойствами.*

Ключевые слова: *симметричных криптоалгоритм; нелинейный узел замен; нелинейность; автокорреляция; спектральное преобразование; криптографическая функция; имитация отжига; ценовая функция.*

1 Введение

Регулярные нелинейные криптографические функции (*узлы замен, блоки подстановок, S-box*) симметричных шифров реализуют отображение n-битных блоков входных данных в m-битные выходные блоки: $F : GF^n(2) \to GF^m(2)$. Традиционный подход к описанию, генерации и оценке показателей стойкости и регулярных S-блоков состоит в представлении функции F с помощью ее координатных функций, которые задаются в терминах булевой алгебры [1]. В тоже время, как показано в [2, 3], построение нелинейных узлов замен с высокими показателями стойкости через итеративное формирование компонентных булевых функций является непрактичным уже при $n = 6$ и вычислительно недостижимым для $n > 6$. Это предполагает обоснование новых подходов к математическому описанию криптографических узлов замен симметричных шифров, исследованию и построению вычислительно эффективных алгоритмов генерации.

В первой части данной работы рассмотрен традиционный способ описания S-блоков, через совокупность компонентных булевых функций. Далее будут использованы недвоичные функции, с помощью которых удается компактно представить S-блоки и формализовать способ их генерации. Авторским коллективом приведены некоторые результаты, касающиеся вычислительного поиска S-блоков с использованием предлагаемого подхода. Результаты данной работы являются логическим продолжением ранее опубликованных исследований [3]. Полученные данные дают основание утверждать, что формируемые S-блоки, по показателям нелинейности и автокорреляции, обладают улучшенными свойствами.

2 Представление S-блоков через компонентные булевы функции

Основные понятия и определения математического аппарата булевой алгебры, используемые при описании нелинейных узлов замен через компонентные булевы функции и оценке

их криптографических свойств, представлены в многочисленных работах [2-6 и др.]. Приведем здесь лишь краткие сведения, основные понятия и определения, которые понадобятся для дальнейшего изложения материала.

Булевой функцией $f(x_1,...,x_n)$ от n переменных является функция, осуществляющая отображение из поля $GF(2^n)$ всех двоичных векторов $x=(x_1,...,x_n)$ длины n в поле $GF(2)$ [4]. Обычно булевы функции представляются в алгебраической нормальной форме (АНФ):

$$f(x_1,...,x_n)=\lambda_0+\lambda_1x_1+...+\lambda_nx_n++\lambda_{12}x_1x_2+\lambda_{13}x_1x_3+...+\lambda_{12...n}x_1x_2...x_n, \quad (1)$$

где $\lambda_0,\lambda_1,\ldots,\lambda_{12},\ldots,\lambda_{12...n}$ - уникальные двоичные константы, а суммирование и умножение производится в двоичном поле GF(2).

Поле $GF(2^n)$ состоит из 2^n векторов $\alpha_i=(\alpha_1^i,\alpha_2^i,...,\alpha_n^i)$, $\alpha_j^i\in GF(2)$: $\alpha_0=(0,...,0,0)$, $\alpha_1=(0,...,0,1)$, …, $\alpha_{2^n-1}=(1,...,1,1)$, $\alpha_i\in V_n$, где V_n – векторное пространство над $GF(2)$.

Таблицей истинности функции f называется (0,1)-последовательность, определенная как [5]:

$$\left(f(x)\middle|x\in GF^n(2)\right)=\left(f(\alpha_0),f(\alpha_1),...,f(\alpha_{2^n-1})\right).$$

Последовательностью функции f, обозначаемой $\hat{f}$, называется (1,-1)- последовательность, определенная как [5]:

$$\left((-1)^{f(x)}\middle|x\in GF^n(2)\right)=\left((-1)^{f(\alpha_0)},(-1)^{f(\alpha_1)},...,(-1)^{f(\alpha_{2^n-1})}\right)$$

Рассмотрим криптографические свойства функций, реализующих отображения из $GF^n(2)$ в $GF^m(2)$, где $1\le m\le n$. Пусть M_n^m есть множество таких функций, а B_n есть множество булевых функций от n переменных, то есть функций, реализующих отображения из $GF^n(2)$ в $GF(2)$. Тогда любую функцию $F\in M_n^m$ можно рассматривать как состоящую из m булевых функций от n переменных, т.е. m-выходных координатных функций из B_n.

В более общем представлении, компонентная функция $F\in M_n^m$ является ненулевой линейной комбинацией ее координатных функций из B_n.

Таким образом, функцию $F:GF^n(2)\to GF^m(2)$ запишем через множество

$$F=(f_1(x_1,...,x_n),...,f_m(x_1,...,x_n)), \text{ где } f_i(x_1,...,x_n)\in B_n.$$

Алгебраическая степень функции f [5], обозначаемая $\deg(f)$, определяется как максимальная степень многочлена, представленного в АНФ.

Важные свойства булевых функций изучаются с использованием преобразования Уолша-Адамара.

Преобразование Уолша-Адамара функции $f(x_1,...,x_n)\in B_n$ есть вещественная функция $\overline{F}(w)$ [5]:

$$\overline{F}(w)=\sum_{x\in GF^n(2)}(-1)^{f(x)+w\cdot x}, \quad (2)$$

где скалярное произведение векторов x и w определяется как $x\cdot w=x_1w_1+...+x_nw_n$.

Булева функция f сбалансирована, если вероятности событий $f(x)=1$ и $f(x)=0$ равны. Используя преобразование Уолша-Адамара, условие сбалансированности функции f запишем в виде $\overline{F}(0)=0$.

Расстояние по Хеммингу между двумя функциями f и g из B_n определяется как:

$$d_H(f,g) = card\left\{x \mid f(x) \neq g(x), x \in GF^n(2)\right\}. \quad (3)$$

Нелинейность $NL(f)$ функции $f(x_1,...,x_n) \in B_n$ определяется как [5]:

$$NL(f) = \min_{g \in A_n} d_H(f,g), \quad (4)$$

где A_n - множество всех аффинных функций от *n* переменных,

$$A_n = \left\{ a_0 + \sum_{i=1}^{n} a_i x_i \mid a_i \in GF(2), 0 \le i \le n \right\}. \quad (5)$$

С использованием преобразования Уолша-Адамара нелинейность функции *f* может быть получена следующим образом:

$$NL(f) = 2^{n-1} - \frac{1}{2} \max_{w \in GF^n(2)} \left|\overline{F}(w)\right|. \quad (6)$$

Взаимосвязь показателя нелинейности $NL(f)$ функции $f(x_1,...,x_n) \in B_n$ с преобразованием Уолша-Адамара и вывод формулы (6) легко понять, представив выражение (2) в виде матричного умножения последовательности функции $\left((-1)^{f(x)} \middle| x \in GF^n(2)\right)$, на матрицу Уолша-Адамара H_{2^n} порядка 2^n:

$$\left(\overline{F}(w) \middle| w \in GF^n(2)\right) = \left(\sum_{x \in F_2^n} (-1)^{f(x)+w \cdot x} \middle| w \in GF^n(2) \right) = \left((-1)^{f(x)} \middle| x \in GF^n(2)\right) \cdot H_{2^n}$$

(*последовательность функции в данном выражении и далее по тексту представляется в виде вектора-строки, образованной элементами этой последовательности*).

Итеративное правило построения матрицы H_{2^n} задается следующим выражением:

$$H_1 = |1|, \; H_{2^i} = \begin{vmatrix} H_{2^{i-1}} & H_{2^{i-1}} \\ H_{2^{i-1}} & -H_{2^{i-1}} \end{vmatrix}, \; i \in N.$$

Каждая строка матрицы Уолша-Адамара соответствует последовательности некоторой аффинной функции $g_i(x_1,...,x_n)$ из A_n с $a_0 = 0$ в общем представлении (5). Полное множество последовательностей всех аффинных функций с $a_0 = 0$ упорядочены по строкам (*столбцам*) матрицы Уолша-Адамара естественным образом:

$$A'_n = \left\{ \begin{array}{c} g_1(x_1,...,x_n) = 0 \\ g_2(x_1,...,x_n) = x_1 \\ g_3(x_1,...,x_n) = x_2 \\ ... \\ g_{2^n}(x_1,...,x_n) = x_1 + x_2 + ... + x_n \end{array} \right\}$$

где $g_i(x_1,...,x_n)$ - *i*-я аффинная функция, из упорядоченного подмножества аффинных функций A'_n с $a_0 = 0$ в (5).

Другими словами, последовательность $\left((-1)^{g_i(x)} \middle| x \in GF^n(2)\right)$ i-й аффинной функции из A'_n соответствует i-й строке матрицы Уолша-Адамара и наоборот.

Тогда, очевидно, выполняется равенство

$$\left((-1)^{f(x)}\middle|x\in GF(2)^n\right)\cdot H_{2^n}=\left((-1)^{f(x)}\middle|x\in GF^n(2)\right)\cdot\begin{pmatrix}\left((-1)^{g_1(x)}\middle|x\in GF^n(2)\right)\\ \left((-1)^{g_2(x)}\middle|x\in GF^n(2)\right)\\ \dots\\ \left((-1)^{g_{2^n}(x)}\middle|x\in GF^n(2)\right)\end{pmatrix}^T=$$

$$=\left(\sum_{x\in GF^n(2)}(-1)^{f(x)+g_i(x)}\middle|x\in GF^n(2),\ g_i(x)\in A'_n\right)=\left(\overline{F}(w)\middle|w\in GF^n(2)\right).$$

Например, для $n = 2$ имеем матрицу Уолша-Адамара H_4:

$$H_4=\begin{vmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{vmatrix}$$

причем

$$\left((-1)^{g_1(x)=0}\middle|x\in GF^2(2)\right)=(1,1,1,1);\ \left((-1)^{g_2(x)=x_1}\middle|x\in GF^2(2)\right)=(1,-1,1,-1);$$

$$\left((-1)^{g_3(x)=x_2}\,x\in GF^2(2)\right)=(1,1,-1,-1);\ \left((-1)^{g_4(x)=x_1+x_2}\middle|x\in GF^2(2)\right)=(1,1,-1,-1)$$

и матричное произведение $\left((-1)^{f(x)}\middle|x\in GF^2(2)\right)\cdot H_4$ соответствует вычислению вектора значений функции $\overline{F}(w)$ для всех $w\in GF^2(2)$.

Выражение для расчета значений коэффициентов преобразования Уолша-Адамара запишем, соответственно, в виде

$$\overline{F}(w)=\sum_{x\in GF^n(2)}(-1)^{f(x)+g_w(x)}\middle|x\in GF^n(2)$$

Максимальное значение коэффициентов преобразования Уолша-Адамара $\max\limits_{w\in GF^n(2)}\overline{F}(w)$ булевой функции *f(x)* соответствует максимальному коэффициенту корреляции (*похожести*) последовательности этой функции и последовательностей всех аффинных функций из множества A'_n:

$$\max_{w\in GF^n(2)}\overline{F}(w)=\max_{w\in GF^n(2)}\left(\sum_{x\in GF^n(2)}(-1)^{f(x)+g_w(x)}\middle|x\in GF^n(2)\right)=$$

$$=\max_{w\in GF^n(2)}\begin{pmatrix}card\left\{x\,|\,f(x)=g_w(x),x\in GF^n(2)\right\}-\\-card\left\{x\,|\,f(x)\neq g_w(x),x\in GF^n(2)\right\}\end{pmatrix}.$$

Последовательности аффинных функций с $a_0=1$ в (5) соответствуют инверсии (*умножению на «-1»*) последовательностей функций из A'_n, следовательно, максимум модуля

коэффициентов преобразования Уолша-Адамара $\max\limits_{w\in GF^n(2)}\left|\overline{F}(w)\right|$ булевой функции *f(x)* будет соответствовать максимальному коэффициенту корреляции последовательности этой функции и последовательностей всех аффинных функций из множества A_n.

По определению нелинейности из (4) имеем:

$$NL(f)=\min_{g\in A_n} d_H(f,g)=\min_{g\in A_n} card\left\{x\mid f(x)\neq g(x), x\in GF^n(2)\right\}.$$

Поскольку

$$card\left\{x\mid f(x)=g(x), x\in GF^n(2)\right\}+card\left\{x\mid f(x)\neq g(x), x\in GF^n(2)\right\}=2^n,$$

то справедливо следующее равенство

$$\max_{w\in GF^n(2)}\left|\overline{F}(w)\right|=\max_{w\in GF^n(2)}\left|\sum_{x\in GF^n(2)}(-1)^{f(x)+g_w(x)}\middle| x\in GF^n(2)\right|=$$

$$=\max_{w\in GF^n(2)}\left|\begin{matrix}card\left\{x\mid f(x)=g_w(x), x\in GF^n(2)\right\}-\\ -card\left\{x\mid f(x)\neq g_w(x), x\in GF^n(2)\right\}\end{matrix}\right|=\max_{w\in GF^n(2)}\left|2^n-2card\left\{x\mid f(x)\neq g(x), x\in GF^n(2)\right\}\right|=$$

$$=2^n-2\min_{g\in A_n} card\left\{x\mid f(x)\neq g(x), x\in GF^n(2)\right\}=2^n-2\min_{g\in A_n} d_H(f,g)=2^n-2N(f),$$

откуда получаем:

$$NL(f)=\frac{2^n-\max\limits_{w\in GF^n(2)}\left|\overline{F}(w)\right|}{2}=2^{n-1}-\max_{w\in GF^n(2)}\left|\overline{F}(w)\right|.$$

Автокорреляционная функция, обозначаемая как $r_{\hat{f}}(\alpha)$, вычисляется по формуле [6]:

$$r_{\hat{f}}(\alpha)=\sum_{x\in GF^n(2)}\hat{f}(x)\hat{f}(x\oplus\alpha)$$

где $\alpha\in GF^n(2)$ и $r_{\hat{f}}(0)=2^n$. Автокорреляционная функция является вектором, содержащим 2^n действительных значений в диапазоне $[(-2)^n,2^n]$.

Автокорреляция AC функции f является максимальным абсолютным значением автокорреляционной функции [6]:

$$AC=\max_{\alpha\in GF^n(2),\alpha\neq 0}\left|r(\alpha)\right|$$

Таким образом, математический аппарат булевых функций является удобным инструментом для описания регулярных нелинейных узлов замен. При этом использование преобразования Уолша-Адамара дает адекватный механизм оценки основных криптографических показателей стойкости, в частности, нелинейности компонентных булевых функций.

В то же время, использование рассмотренного математического аппарата для генерации регулярных узлов замен через итеративное формирование компонентных булевых функций, является непрактичным уже при $n = 6$ и вычислительно недостижимым для $n > 6$ [2, 3].

Перспективным направлением в этом смысле является использование недвоичных криптографических функций, описывающих отображение *n*-битных блоков входных данных в *m*-битные выходные блоки в нелинейном узле замен в виде функций отображения $F: GF(2^n)\to GF(2^m)$.

3 Представление S-блоков через недвоичные булевы функции

Введем основные понятия и определения для описания нелинейных узлов замен через недвоичные функции.

Недвоичной (над полем $GF(2^{n_1})$) *функцией* $F(X_1,...,X_{n_2})$ от n_2 переменных является функция, осуществляющая отображение из поля $GF((2^{n_1})^{n_2})$ всех векторов $X=(X_1,...,X_{n_2})$ длины n_2 с элементами из $GF(2^{n_1})$ в поле $GF(2^{n_1})$. Как и рассмотренные выше булевы функции, каждая недвоичная функция $F(X_1,...,X_{n_2})$ может быть представлена в АНФ, т.е. как сумма произведений составляющих координат:

$$F(X_1,...,X_{n_2})=\Lambda_0+\Lambda_1X_1+...+\Lambda_nX_n+\Lambda_{12}X_1X_2+\Lambda_{13}X_1X_3+...+\Lambda_{12...n_2}X_1X_2...X_n, \quad (7)$$

где $\Lambda_0,\Lambda_1,...,\Lambda_{12},...,\Lambda_{12...n_2}$ - уникальные константы из $GF(2^{n_1})$, суммирование и умножение также производится в поле $GF(2^{n_1})$.

Поле $GF((2^{n_1})^{n_2})$ состоит из $2^{n_1n_2}$ векторов $\mathrm{A}_i=(\mathrm{A}_1^i,\mathrm{A}_2^i,...,\mathrm{A}_{n_2}^i)$, $\mathrm{A}_j^i\in GF(2^{n_1})$:

$$\mathrm{A}_0=(0,...,0,0),\ \mathrm{A}_1=(0,...,0,1),\ \ldots,\ \mathrm{A}_{2^{n_1}-1}=(0,...,0,2^{n_1}-1),\ \mathrm{A}_{2^{n_1}}=(0,...,1,0),$$
$$\mathrm{A}_{2^{n_1}+1}=(0,...,1,1),\ \ldots,\ \mathrm{A}_{(2^{n_1})^{n_2}-1}=(2^{n_1}-1,...,2^{n_1}-1,2^{n_1}-1),\ \alpha_i\in V_n,$$

где V_{n_2} – векторное пространство над $GF(2^{n_1})$.

Поле $GF((2^{n_1})^{n_2})$ изоморфно полю $GF(2^n)$, $n=n_1n_2$, т.е. имеем взаимно-однозначное функциональное соответствие множества векторов $\mathrm{A}_i=(\mathrm{A}_1^i,\mathrm{A}_2^i,...,\mathrm{A}_{n_2}^i)\in V_{n_2}$ с элементами из $GF(2^{n_1})$ и двоичных векторов $\alpha_i=(\alpha_1^i,\alpha_2^i,...,\alpha_n^i)\in V_n$.

Таблицей истинности недвоичной (*над полем* $GF(2^{n_1})$) функции *F* называется последовательность с элементами из $GF(2^{n_1})$, определенная как:

$$\left(F(X)\middle|X\in GF^{n_2}(2^{n_1})\right)=\left(F(\mathrm{A}_0),F(\mathrm{A}_1),...,F(\mathrm{A}_{2^{n_1n_2}-1})\right).$$

Последовательностью недвоичной (над полем $GF(2^{n_1})$) функции *F* называется последовательность из $2^{n_1n_2}$ (1,-1)-кортежей длины $2^{n_1}-1$ каждый, определенная как:

$$\left((-1)^{w\cdot F(X)}\middle|w\in GF(2^{n_1}),w\neq 0,X\in GF^{n_2}(2^{n_1})\right)=$$
$$=\begin{pmatrix}\left((-1)^{w_1\cdot F(\mathrm{A}_0)},(-1)^{w_2\cdot F(\mathrm{A}_0)},...,(-1)^{w_{2^{n_1}-1}\cdot F(\mathrm{A}_0)}\right),\\ \left((-1)^{w_1\cdot F(\mathrm{A}_1)},(-1)^{w_2\cdot F(\mathrm{A}_1)},...,(-1)^{w_{2^{n_1}-1}\cdot F(\mathrm{A}_1)}\right),...,\\ \left((-1)^{w_1\cdot F(\mathrm{A}_{2^{n_1n_2}})},(-1)^{w_2\cdot F(\mathrm{A}_{2^{n_1n_2}})},...,\right.\\ \left.(-1)^{w_{2^{n_1}-1}\cdot F(\mathrm{A}_{2^{n_1n_2}})}\right)\end{pmatrix},$$

где w – накладываемая маска и $w \cdot F(X)$ - скалярное произведение векторных представлений чисел w и $F(X) \in GF(2^{n_1})$ (т.е. числа представлены в виде двоичных последовательностей, с элементами из $GF^{n_1}(2)$).

Например, пусть $n_1 = 2$, $n_2 = 1$ и недвоичная (над $GF(2^2)$) функция задана в АНФ следующим образом: - $F(X) = 3 + X + 2X^2$, где коэффициенты многочлена принадлежат полю $GF(2^2)$: $0 = (0,0)$, $1 = (1,0)$, $2 = (0,1)$, $3 = (1,1)$.

Входными элементами такой функции являются однокоординатные вектора (*скаляры*) с элементами из $GF(2^2)$: $A_0 = (0)$, $A_1 = (1)$, $A_2 = (2)$, $A_3 = (3)$.

Таблицей истинности функции $F(X) = 3 + X + 2X^2$ является последовательность с элементами из $GF(2^2)$:

$$\left(F(X)\middle| X \in GF^1(2^2)\right) = \left(F(A_0), F(A_1), F(A_2), F(A_3)\right) = (3, 0, 0, 3).$$

Значения маски w и выхода функции $F(X)$ принадлежат полю $GF(2^2)$: $0 = (0,0)$, $1 = (1,0)$, $2 = (0,1)$, $3 = (1,1)$.

Последовательностью функции $F(X) = 3 + X + 2X^2$ является последовательность из $2^{n_1 n_2} = 4$ (1,-1)-кортежей с длинной каждого $2^{n_1} - 1 = 3$:

$$\left(\begin{array}{l}\left((-1)^{w_1 \cdot F(X)}, (-1)^{w_2 \cdot F(X)}, (-1)^{w_3 \cdot F(X)}\right) \\ \left| w \in GF(2^{n_1}), w \neq 0, X \in GF^1(2^2)\right.\end{array}\right) = \left(\begin{array}{l}\left((-1)^{w_1 \cdot F(A_0)}, (-1)^{w_2 \cdot F(A_0)}, (-1)^{w_3 \cdot F(A_0)}\right), \\ \left((-1)^{w_1 \cdot F(A_1)}, (-1)^{w_2 \cdot F(A_1)}, (-1)^{w_3 \cdot F(A_1)}\right), \\ \left((-1)^{w_1 \cdot F(A_2)}, (-1)^{w_2 \cdot F(A_2)}, (-1)^{w_3 \cdot F(A_2)}\right), \\ \left((-1)^{w_1 \cdot F(A_3)}, (-1)^{w_2 \cdot F(A_3)}, (-1)^{w_3 \cdot F(A_3)}\right)\end{array}\right) =$$

$$= \left(\begin{array}{l}\left((-1)^{(1,0)\cdot(1,1)}, (-1)^{(0,1)\cdot(1,1)}, (-1)^{(1,1)\cdot(1,1)}\right), \\ \left((-1)^{(1,0)\cdot(0,0)}, (-1)^{(0,1)\cdot(0,0)}, (-1)^{(1,1)\cdot(0,0)}\right), \\ \left((-1)^{(1,0)\cdot(0,0)}, (-1)^{(0,1)\cdot(0,0)}, (-1)^{(1,1)\cdot(0,0)}\right), \\ \left((-1)^{(1,0)\cdot(1,1)}, (-1)^{(0,1)\cdot(1,1)}, (-1)^{(1,1)\cdot(1,1)}\right)\end{array}\right) = \left(\begin{array}{l}\left((-1)^1, (-1)^1, (-1)^0\right), \left((-1)^0, (-1)^0, (-1)^0\right), \\ \left((-1)^0, (-1)^0, (-1)^0\right), \left((-1)^1, (-1)^1, (-1)^0\right)\end{array}\right) =$$

$$= \left((-1,-1,1), (1,1,1), (1,1,1), (-1,-1,1)\right).$$

Рассмотрим криптографические свойства функций F', реализующих отображения из $GF^{n_2}(2^{n_1})$ в $GF^m(2^{n_1})$, где $1 \le m \le n_2$.

Пусть $M_{n_2}^m$ есть множество таких функций F', а B_{n_2} - это множество недвоичных функций $F(X_1, ..., X_{n_2})$ от n_2 переменных, т.е. функций, реализующих отображения из $GF^{n_2}(2^{n_1})$ в $GF(2^{n_1})$.

Тогда любую функцию из $M_{n_2}^m$ можно рассматривать как состоящую из m недвоичных функций $F(X_1, ..., X_{n_2})$ от n_2 переменных, т.е. m-выходных координатных функции из B_{n_2}.

В более общем представлении, компонентная функция из $M_{n_2}^m$ является ненулевой линейной комбинацией ее координатных недвоичных функций из B_{n_2}.

Таким образом, функцию отображения $F': GF^{n_2}(2^{n_1}) \to GF^m(2^{n_1})$, реализующую нелинейный узел замен, запишем через множество $F' = (F_1(X_1,...,X_{n_2}),...,F_m(X_1,...,X_{n_2}))$, где $F_i(X_1,...,X_{n_2}) \in B_{n_2}$.

В данной работе мы ограничимся лишь рассмотрением функций с m = 1, т.е. будем рассматривать только функции $F' = F_i(X_1,...,X_{n_2})$, реализующие отображения из $GF^{n_2}(2^{n_1})$ в $GF(2^{n_1})$.

Введенная ранее формализация является естественным обобщением рассмотренного выше подхода к *представлению регулярных узлов замен в виде совокупности компонентных булевых функций*. Действительно, используя традиционный подход к описанию функционального отображения n-битных блоков входных данных в m-битные выходные блоки функцию $F: GF^n(2) \to GF^m(2)$, где $n = n_1 n_2$, $m = n_1$, можно представить в виде кортежа $F = \{ f_1(x_1,...,x_n), f_2(x_1,...,x_n),..., f_m(x_1,...,x_n) \}$ из $m = n_1$ булевых функций от $n = n_1 n_2$ булевых переменных каждая.

Для недвоичной функции, из предыдущего примера, имеем следующее соответствие:

$$F(X) = 3 + X + 2X^2 \equiv \left\{ \begin{array}{l} f_1(x_1, x_2) = 1 + x_1 + x_2 \\ f_2(x_1, x_2) = 1 + x_1 + x_2 \end{array} \right\},$$

где знак тождества означает тождественность правила отображения n = *2*-битных блоков входных данных в m = *2*-битные выходные блоки.

Важные свойства булевых функций изучаются с использованием преобразования Уолша-Адамара. По аналогии с преобразованием Уолша-Адамара введем спектральное преобразование недвоичных функций следующим образом.

Поскольку нелинейность булевой функции определяется как минимальное расстояние по Хэммингу от рассматриваемой функции ко множеству всех аффинных булевых функций, для недвоичного случая нам необходимо определить множество аффинных недвоичных функций.

Алгебраическая степень F, обозначаемая $\deg(F)$, определяется как максимальная степень многочлена, представленного в АНФ.

По аналогии с классическим подходом назовем аффинной функцией недвоичную функцию F, чья алгебраическая степень $\deg(F) \le 1$. Соответственно, недвоичные функции, имеющие алгебраическую степень $\deg(F) > 1$, назовем нелинейными недвоичными функциями.

Спектральным преобразованием недвоичной функции $F(X_1,...,X_{n_2}) \in B_{n_2}$ есть вещественная функция $\overline{F}(W)$:

$$\overline{F}(W) = \sum_{X \in GF^{n_2}(2^{n_1})} \sum_{i=1}^{n_1} (-1)^{(F(X))_i + (G_W(x))_i}, \quad (8)$$

где, под $G_W(x)$ - понимается W-я недвоичная аффинная функция от n_2 переменных из множества A_n:

$$A_n = \left\{ a_0 + \sum_{i=1}^{n_2} a_i X_i \mid a_i \in GF(2^{n_1}), 0 \le i \le n \right\}. \quad (9)$$

Также как и вектор *w*, в случае булевого описания, определяет вид линейных двоичных функций $g_w(x)$, в случае недвоичного описания вектор *W* задает вид недвоичных аффинных функций $G_W(x)$.

4 Вычислительный метод генерации регулярных нелинейных узлов замен

На сегодняшний день известно большое число различных вычислительных техник генерации регулярных нелинейных узлов замен. Среди наиболее известных из них следует выделить [2, 8-12]: - побитовые методы (*bit-by-bit methods*); - методы случайной генерации с фильтрацией (*random generation*); - метод градиентного подъема (*hill climbing*); - генетические алгоритмы (*genetic algorithms*); - метод имитации отжига (*simulated annealing*); - метод дифференциальной эволюции (*differential evolution*); - метод оптимизации роем частиц (*particle swarm optimization*) и др.

Одним из наиболее эффективных является метод имитации отжига SA. Так, например, в [10, 13] показано, что использование метода имитации отжига позволяет формировать криптографические функции с очень высокими показателями стойкости. Приведем краткое изложение этого метода из [10].

Поиск начинается с некоторого начального состояния $S = S_0$. Параметр T – некий контрольный параметр, известный как температура. T инициализируется высокой температурой T_0 и постепенно снижается. При каждом значении температуры, выполняется определенное число *MIL* (*Moves in Inner Loop*, шагов во внутреннем цикле) шагов к новым состояниям. Состояние-кандидата *Y* выбирается случайным образом из соседей *N*(*S*) текущего состояния. Вычисляется изменение значения функции *cost*, δ = *cost*(*Y*) – *cost*(*S*). Если значение *cost*(*S*) улучшается (*т.е.* $\delta < 0$ *для задачи минимизации*), тогда выполняется шаг относительно этого состояния (*S* = *Y*); в противном случае – он выполняется с некоторой вероятностью. Т.о. чем хуже шаг, тем меньше вероятность того, что он будет принят; чем ниже температура *T*, тем менее вероятно, что ухудшающий шаг будет принят. Вероятностное принятие решения определяется генерацией случайного числа *U* в интервале (0…1) и выполнением указанного ниже сравнения.

Вначале температура высокая и принимается почти каждый шаг. Это сделано для того, чтобы поиск носил не локальный, а глобальный характер. По мере того как температура уменьшается, становится все более трудно принимать ухудшающие шаги. В конце концов, допускаются только улучшающие шаги и процесс застывает. Алгоритм прерывается, когда встречается критерий остановки. Общий критерий остановки (*который и был применен в нашей работе*) – остановка поиска при достижении заданного числа *MaxIL* внутренних циклов, либо когда было выполнено некоторое максимальное число *MUL* последовательных непродуктивных внутренних циклов (*т.е. без единого принятого шага*). При этом лучшее достигнутое состояние сохраняется, поскольку поиск может выйти из него и впоследствии не найти состояние с подобными показателями. В конце каждого внутреннего цикла температура понижается. В данной работе использовалось ***геометрическое охлаждение*** *– умножение на константу охлаждения* α *в интервале (0…1)*.

Соседей функции f можно определить следующим образом. Функция *g* находится по соседству с функцией *f*, если:

$$\exists x, y \in Z_2^n : \hat{f}(x) \neq \hat{f}(y), \hat{g}(x) = \hat{f}(y), \hat{g}(y) = \hat{f}(x), \forall z \in Z_2^n \setminus \{x, y\} : \hat{g}(z) = \hat{f}(x).$$

Алгоритм имитации отжига SA

```
S = S0;
T = T0;
repeat {
   for (int i = 0; i < MIL; i++)
   {
```

выбрать $Y \in N(S)$;
$\delta = cost(Y) - cost(S)$;
if ($\delta < 0$) **then** $S = Y$;
else сгенерировать $U = U(0,1)$;
if ($U < exp(-\delta / T)$) **then** $S = Y$;
}
$T = T \times \alpha$;
}
until (критерий остановки не достигнут).

Поиск начинался со сбалансированной, но при этом случайной функции. Один шаг алгоритма меняет местами два отличных элемента таблицы истинности функции, сохраняя ее сбалансированность.

Рассмотрим процедуры формирования функций стоимости cost (*ценовых функций*), используемые для генерации S-блоков через спектральные характеристики булевых функций, введем соответствующие функции стоимости для генерации S-блоков через спектры недвоичных криптографических функций.

Пусть функция $F(x): GF^n(2) \to GF^m(2)$ задает S-блок размерности *nxm*. Пусть для $\beta \in GF^m(2)$, $F_\beta(x) = \beta_1 f_1(x) \oplus ... \oplus \beta_m f_m(x)$ - линейная комбинация *m* выходов S-блока *F*. Тогда $\hat{F}_\beta(\omega), r_\beta(s)$ - значения преобразования Уолша-Адамара и значения автокорреляции для каждой булевой функции f_β.

Поскольку нелинейность булевой функции $NL(f) = \frac{1}{2}(2^n - \max_\omega |\hat{F}(\omega)|)$, то задача повышения нелинейности может быть представлена, как задача минимизации абсолютного максимального значения коэффициента Уолша-Адамара.

Изначально в задачах генерации S-блоков для метода имитации отжига использовалась следующая функция стоимости [10]:

$$cost(f) = WHT_{\max}(f) = \max_\omega |\hat{F}(\omega)|.$$

Поскольку задача понижения автокорреляции представляется как ***задача минимизации максимального значения автокорреляционной функции***, то *cost* функция в дальнейших исследованиях приняла следующий вид [10]:

$$cost(f) = AC(f) = \max_{s \neq 0} |\sum_x \hat{f}(x)\hat{f}(x \oplus s)| = \max_{s \neq 0} |\hat{r}(s)|.$$

Обычно в многокритериальных задачах применяется следующий подход: - вычисляется сумма отдельных *cost* функций (*по различным критериям*), умноженных на весовые коэффициенты. Тогда *cost* функция в задаче генерацции S-блока с высокой нелинейностью и низкой автокорреляцией принимает вид [10]:

$$cost(f) = \alpha \cdot WHT_{\max}(f) + \beta \cdot AC(f).$$

Далее были разработаны *улучшенные* функции, которые основывались на следующем положении. Известно, что равенство Парсеваля $\sum_w (\hat{F}(w))^2 = 2^{2n}$ ограничивает $WHT_{\max}(f) = \max_w |\hat{F}(w)|$ значением равным как минимум $2^{n/2}$.

Данная граница достигается тогда, когда выполняется равенство $|\hat{F}(w)| = 2^{\frac{n}{2}}$ для каждого ω. Когда значение некоторого коэффициента $|\hat{F}(w)|$ меньше этой идеальной границы,

теорема Парсеваля утверждает, что другие значения коэффициентов $\left|\hat{F}(w)\right|$ должны быть выше этой границы. Таким образом, попытка ограничить отдаленность абсолютных значений коэффициентов Уолша-Адамара от данной границы является возможным средством достижения высокой нелинейности. Спектры некоторых функций содержат все значения (*по модулю*), равные этой идеальной границы. Такие функции называются бент-функциями.

Помимо обладания наивысшей возможной нелинейностью эти функции имеют нулевую автокорреляцию. Следовательно, функция стоимости

$$cost(f) = \sum_{\omega \in GF^n(2)} \left| \left|\hat{F}(\omega)\right| - 2^{\frac{n}{2}} \right|^R \tag{10}$$

является возможным подходом к оптимизации нелинейности и автокорреляции. В виду несбалансированности бент-функций приведенная функция стоимости *cost* может быть улучшена для нахождения сбалансированных криптографических функций. В [10] было введено обобщение функции стоимости (10), которое приняло следующий вид:

$$cost(f) = \sum_{\omega \in GF^n(2)} \left| \left|\hat{F}(\omega)\right| - X \right|^R . \tag{11}$$

Параметры X и R, называемые *весовыми коэффициентами*, обеспечивают свободу для экспериментирования и поиска оптимальных значений.

По аналогии с функциями стоимости относительно спектра Уолша-Адамара вида (11), функции стоимости относительно спектра автокорреляционной функции имеют следующий вид:

$$cost(f) = \sum_{s \in GF^n(2)} \left| \left|r(s)\right| - X \right|^R . \tag{12}$$

Традиционно, ценовые функции применяются для *оптимизации* отдельной булевой функции. Для всего же нелинейного узла замен *cost* функции, основанные на спектре Уолша-Адамара, можно обобщить следующим образом [10]:

$$cost(f) = \sum_{\beta \in GF^m(2)} \sum_{\omega \in GF^n(2)} \left| \left|\hat{F}_\beta(\omega)\right| - X \right|^R \tag{13}$$

и аналогично для *cost* функций, основанных на автокорреляционном спектре:

$$cost(f) = \sum_{\beta \in GF^m(2)} \sum_{s \in GF^n(2)} \left| \left|r_\beta(s)\right| - X \right|^R . \tag{14}$$

Для оптимизации по критериям *нелинейности* и *автокорреляции* в [13] использовалась следующая функция стоимости:

$$cost(f) = \sum_{\beta \in GF^m(2)} \sum_{\omega \in GF^n(2)} \left| \left|\hat{F}_\beta(\omega)\right| - X_1 \right|^{R_1} + \sum_{\beta \in GF^m(2)} \sum_{s \in GF^n(2)} \left| \left|r_\beta(s)\right| - X_2 \right|^{R_2} . \tag{15}$$

В первой части исследований, проводимых в рамках данной работы, использовались функции стоимости вида (12), (13), с заменой ***спектральных коэффициентов Уолша-Адамара*** и ***коэффициентов автокорреляционных спектров булевых функций*** на предложенные выше ***коэффициенты соответствующих спектров недвоичных функций***.

Вторая часть исследований состояла в совершенствовании функций стоимости (*критерия поиска криптографических функций*), базирующемся на изложенном ниже положении.

Известно, что при оптимизации криптографической функции по нелинейности и автокорреляции, она по своим спектральным характеристикам (*спектру корреляции с линейными функциями и автокорреляционному спектру*) стремится к спектральным характеристикам бент-функций, что и было использовано в предыдущих работах [10,13] при разработке функций вида (10)-(15). В тоже время, очевидным недостатком такого подхода, является использование одного (*фиксированного*) значения коэффициента, к которому стремятся все спектральные значения оптимизируемой криптографической функции. При этом значения спектральных коэффициентов идеальной функции (*или бент-функции*) состоят из двух возможных значений для булевых функций, и из трех значений для введенных недвоичных функций. При введении же дополнительных ограничения на сбалансированность, количество возможных значений спектральных коэффициентов еще более возрастает.

При разработке новых функций стоимости, в выражениях (11) - (15) предлагается заменить весовой коэффициент X на т.н. ***динамические весовые коэффициенты***, т.е. весовые коэффициенты, принимающие различные значения для различных входных индексов спектра.

В данной работе в качестве значений ***динамических весовых коэффициентов*** используются *спектральные значения бент-функций*. Предлагаемые функции стоимости имеют вид:

$$cost(f) = \sum_{\omega \in GF^n(2)} \left| \hat{F}(\omega) - \hat{B}(\omega) \right|^R, \tag{16}$$

$$cost(f) = \sum_{s \in GF^n(2)} \left| r_F(s) - r_B(s) \right|^R, \tag{17}$$

$$cost(f) = \sum_{\omega \in GF^n(2)} \left| \hat{F}(\omega) - \hat{B}(\omega) \right|^{R_1} + \sum_{s \in GF^n(2)} \left| r_F(s) - r_B(s) \right|^{R_2}, \tag{18}$$

где $\hat{B}(\omega), r_B(s)$ – спектральные значения нелинейности и автокорреляции случайной недвоичной бент-функции B.

Таким образом, в основу предлагаемого вычислительного метода генерации регулярных нелинейных узлов замен, положено применение недвоичных криптографических функций и усовершенствованных ценовых функций (16) – (18), с использованием динамических весовых коэффициентов $\hat{B}(\omega), r_B(s)$. Усовершенствованный таким образом ***метод имитации отжига*** позволяет (*как показано ниже*) реализовать вычислительный поиск регулярных узлов замен с улучшенными показателями нелинейности и автокорреляции.

5 Результаты экспериментальных исследований

В данной работе представлены результаты экспериментальных исследований эффективности предлагаемого вычислительного метода (*генерации регулярных нелинейных узлов замен*). В этой связи важно подчеркнуть, что 1-я часть исследований была проведена с использованием ***спектров недвоичных функций с весовыми коэффициентами в функциях стоимости (13) – (15) метода имитации отжига***, а вторая часть исследований – ***с использованием динамических коэффициентов*** $\hat{B}(\omega), r_B(s)$ ***в функциях стоимости (16) – (18)***.

Параметры алгоритма для всех исследований были заданы следующим образом:

- $\alpha = 0.95$ – параметр геометрического охлаждения;
- $MIL = 500$ – число шагов, предпринимаемых во внутреннем цикле;
- $MaxIL = 300$ – максимальное число внутренних циклов поиска;
- $MUL = 50$ – максимальное число последовательных непродуктивных внутренних циклов;
- количество пробегов алгоритма для каждого набора параметров равно 10.

В ходе экспериментов с функциями стоимости вида (13) – (15) использовались различные значения статических коэффициентов X и фиксированное значение $R = 3$.

Формировались S-блоки размерностей 8×2, 4×4 и 6×4. Узлы замен выходной размерности 4 представлялись через одну недвоичную функцию над $GF(2^4)$.

Полученные экспериментальные результаты для S-блоков 8×2 приведены ниже, в табл. 1. Лучший полученный результат выделен **жирным шрифтом** на красном фоне.

Таблица 1 – Результаты для S-блоков 8×2

Способ построения спектров, критерий отбора	***Статические коэффициенты***		***Динамические коэффициенты***	
	NL	***AC***	***NL***	***AC***
WHT, ср.	110	56	114	24
WHT, худш.	108	56	**116**	**24**
ACT, ср.	110	48	114	24
ACT, худш.	112	40	114	24
WHT+ACT, худш.	112	40	114	24
WHT+ACT, ср.	112	40	114	24

Как следует из данных Табл. 1, использование функций стоимости с динамическими весовыми коэффициентами, позволило улучшить показатели стойкости нелинейности и автокорреляции формируемых узлов замен.

В табл. 2 проведено сравнение полученных результатов уже с известными результатами, использующими традиционный подход описания S-блока, в виде совокупности компонентных булевых функций [8-12]. Как видно из табл. 2, использование динамических весовых коэффициентов позволило получить лучшие результаты для S-блоков 8×2.

Таблица 2 – Результаты сравнения

Используемый метод генерации	***NL***	***AC***
Случайная генерация	108	56
Генетические алгоритмы	110	48
Имитация отжига (*булевы функции*)	114	32
Имитация отжига (*недвоичные функции*)	112	40
Имитация отжига (*недвоичные функции, динамические весовые коэффициенты*)	**116**	**24**

В табл. 3 приведены лучшие полученные результаты для S-блоков размером 4×4 и 6×4.

Таблица 3 – Лучшие из полученных результатов *(для 4×4 и 6×4)*

Размерность S-блока	***Метод имитации отжига***	***NL***	***AC***
4×4	Булевы функции	4	8
4×4	Функции над $GF(2^4)$	4	8
6×4	Булевы функции	22	24
6×4	Функции над $GF(2^4)$	**24**	**24**

Как следует из приведенной таблицы, применение предлагаемого подхода позволяет повысить нелинейность формируемых S-блоков размерностью 6×4. Подобные S-блоки (*размерности 6×4*) применяются, например, в *DES*-подобных шифрах.

6 Выводы

1. Методы генерации нелинейных узлов симметричных криптоалгоритмов постоянно развиваются и совершенствуются. Одним из перспективных направлений в этом смысле является использование недвоичных криптографических функций.

2. Авторским коллективом рассмотрен один из способов подобной формализации и показано, что использование недвоичных функций позволяет, кроме прочего, интерпретировать известные методы генерации S-блоков. В частности, был реализован метод имитации отжига при недвоичном представлении функций. По результатам экспериментов показано, что в совокупности с улучшенными ценовыми функциями (*настраиваемый элемент метода имитации отжига*) удается сформировать S-блоки с улучшенными показателями стойкости.

3. Предложенный вычислительный метод генерации S-блоков реализован программно.

4. Полученные результаты хорошо согласуются с результатами вычислительных методов традиционного подхода (*через описание S-блока булевыми функциями*).

5. Использование динамических весовых коэффициентов в функциях стоимости, позволили получить лучшие результаты для S-блоков 8×2. Для S-блоков с размерностью 6×4 удалось поднять верхнюю границу показателя нелинейности.

6. Разработанный вычислительный метод предлагается использовать для целей генерации *DES*-подобных S-блоков.

7. Перспективным направлением дальнейших исследований связано с возможностями адаптации предлагаемого подхода для узлов замен больших размерностей и обобщение динамических весовых коэффициентов в функциях стоимости.

Ссылки

[1] *Сорока Л.С.* Вероятностная модель формирования нелинейных узлов замен для симметричных криптографических средств защиты информации / Сорока Л.С., Кузнецов А.А., Московченко И.В., Исаев С.А. // Системи обробки інформації. – Х.:ХУВС, 2009. - № 3 (77). – С. 101-104.

[2] *O'Connor L.* An analysis of a class of algorithms for S-box construction / O'Connor L. // J. Cryptology/. -1994. – P. 133-151.

[3] *Сорока Л.С.* Исследование вероятностных методов формирования нелинейных узлов замен / Сорока Л.С., Кузнецов А.А., Исаев С.А. // Системи обробки інформації. – 2011. - № 8 (98). – С. 113 – 122.

[4] *Булева функция* [Електронний ресурс] // Режим доступу: http://ru.wikipedia.org/wiki/Булева_функция.

[5] *Dawson E.* Designing Boolean functions for cryptographic applications / Dawson E., Millan W., Simpson L. // Contributions to General Algebra, Verlag Johannes Heyn, Klagenfurt. – 2000. – 12. – P. 1-22.

[6] *Clark J.A.* Evolving Boolean functions satisfying multiple criteria / Clark J.A., Jacob J.L., Stepney S., Maitra S., Milan W. // Lecture Notes in Computer Science (2551), Springer, Berlin. – 2002. - 2251. - P. 246-259.

[7] *Parker M.G.* Generalised S-Box Nonlinearity / Parker M.G. // NES/DOC/UIB/WP5/020/A. – 2003.

[8] *Millan W.* How to improve the nonlinearity of bijective s-boxes / Millan W. // Information Security and Privacy, ACISP '98, Springer Verlag. – 1998. – volume 1438 of Lecture Notes in Computer Science. – P. 181-192.

[9] *Millan W.* Evolutionary Heuristics for Finding Cryptographically Strong S-Boxes / Millan W., Burnett L., Carter G., Clark A., Dawson E. // Information and communication security, Springer, Heidelberg. – 1999. – Lecture Notes in Computer Science Volume 1726. – P.263-274.

[10] *Clark J.A.* The Design of S-Boxes by Simulated Annealing / Clark J.A., Jacob J.L., Stepney S. // New Generation Computing. – 2005. – 23(3). – P.219–231.

[11] *Laskari C.* Utilizing Evolutionary Computation Methods for the Design of S-Boxes / Laskari C., Meletiou C., Vrahatis N. // Computational Intelligence and Security. – 2006. – Volume 2. – P.1299-1302.

[12] *Tesar P.* A new method for generating high non-linearity S-Boxes / Tesar P. // Radioengineering. – 2010. - Part I of II, Vol. 19 Issue 1. – P.23 - 26.

[13] *Kavut S.* Improved Cost Function in the Design of Boolean Functions Satisfying Multiple Criteria / Kavut S., Yücel M.D. // Proc. INDOCRYPT. – 2003. – P.121-134.

[14] *Kwangjo K.* Securing DES S-Boxes against Three Robust Cryptanalysis / Kwangjo K., Sangjin L., Sangjoon P., Daiki L. // Proceedings of the Workshop on Selected Areas in Cryptography, SAC '95. – 1995. – P.145-157.

Рецензент: Сергій Толюпа, д.т.н., проф., Київський національний університет імені Т. Шевченка, м. Київ, Україна
E-mail: tolupa@i.ua

Надійшло: Січень 2020.

Автори:

Микита Гончаров, студент факультету комп'ютерних наук, ХНУ імені В.Н. Каразіна, Харків, Україна.
E-mail: worldxdark@gmail.com

Тетяна Кузнецова, науковий співробітник, ХНУ імені В.Н. Каразіна, Харків, Україна.
E-mail: kuznetsova.tatiana17@gmail.com

Олександр Кузнецов, д.т.н., проф., ХНУ імені В.Н. Каразіна, Харків, Україна.
E-mail: kuznetsov@karazin.ua

Недвійкові криптографічні функції для генерації блоків підстановок симетричних шифрів.

Анотація. Важливим елементом сучасних симетричних криптоалгоритмів є нелінійні вузли замін (блоки підстановок, S-блоки). Властивості цих блоків безпосередньо впливають на показники криптографічної стійкості алгоритмів шифрування. Наприклад, стійкість симетричних шифрів до диференціального і лінійного криптоаналізу безпосередньо залежить від показників нелінійності S-блоків та їх кореляційних властивостей. У даній статті досліджуються способи математичного опису регулярних нелінійних вузлів замін, обчислювальні техніки їх генерації і оцінки криптографічних показників. Розглядається традиційний підхід, в якому для опису внутрішньої структури S-блоків використовується сукупність компонентних булевих функцій. Досліджуються недвійкові функції, за допомогою яких формуються регулярні нелінійні вузли заміни. Наводяться результати обчислювального пошуку S-блоків з використанням запропонованого підходу. Для генерації регулярних S-блоків використовувалися методи імітації віджигу, стосовно недвійкових функцій з поліпшеними ціновими функціями (елемент методу імітації віджигу, що настроюється). Показано, що за нелінійності і автокореляції, сформовані вузли замін мають поліпшені властивості.

Ключові слова: симетричний криптоалгоритм; нелінійний вузол замін; не лінійність; автокореляція; спектральне перетворення; криптографічна функція; імітація віджигу; цінова функція.

Reviewer: Serhii Toliupa, Doctor of Sciences (Eng.), Full Prof., Taras Shevchenko National University of Kiev, 81 Lomonosova St., Kyiv, 03189, Ukraine. E-mail: tolupa@i.ua

Received on January 2020.

Authors:
Nikita Goncharov, Computer Science Student, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: worldxdark@gmail.com
Tetiana Kuznetsova, Researcher, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: kuznetsova.tatiana17@gmail.com
Alexandr Kuznetsov, Doctor of Sciences (Eng.), Prof., V. N. Karazin Kharkiv National University, Department of information systems and technologies security, Kharkiv, Ukraine.
E-mail: kuznetsov@karazin.ua

Non-binary cryptographic functions for design of blocks of substitutions of symmetric cipher.

Annotation. In the paper considers the traditional way of describing S-blocks through a set of component Boolean functions.non-binary functions were used to represent s-blocks compactly and formalize the method of generating them. Some results of computational search for S-blocks were presented using the proposed approach, where the generated S-blocks have improved properties based on non-linearity and autocorrelation indicators.

Key words: Symmetric cryptoalgorithm; Nonlinearity replacement nodes; Nonlinearity; Autocorrelation; Spectrum transformation; Cryptographic function; Simulated annealing; Cost function.

ISSN 2519-2310 DOI: 10.26565/2519-2310-2019-4

KhNU

Наукове видання

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА

Випуск 4(16) 2019

Міжнародний електронний науково-теоретичний журнал

Англійською, українською, російською мовами

Комп'ютерне верстання – Федоренко В.В., Єсіна М.В.

61022, Харків, майдан Свободи, 6
Харківський національний університет імені В.Н. Каразіна

V. N. Karazin Kharkiv National University Publishing

2019