DOI 10.31891/2307-5732-2019-269-1-148-154
УДК 004.056.55

А.В. ПРИЙМАК, О.В. САЛІЄВА, Я.Ю. ЯРЕМЧУК
Вінницький національний технічний університет

ДОСЛІДЖЕННЯ МОЖЛИВОСТІ ВИКОРИСТАННЯ АЛГОРИТМУ ЦИКЛІЧНОГО НАДЛИШКОВОГО КОДУ ДЛЯ ПІДВИЩЕННЯ СТІЙКОСТІ КРИПТОСХЕМИ ECIES

В роботі розглянуто проблему вразливості асиметричної криптосхеми ECIES до атаки малими підгрупами, а також досліджено можливість використання алгоритму циклічного надлишкового коду для підвищення її криптостійкості. Розроблено метод перевірки публічного ключа на справжність, який унеможливлює проведення успішної атаки *малими підгрупами і як результат* значно підвищує теоретичну стійкість даного криптоалгоритму. Проведене статистичне тестування запропонованої модифікації алгоритму показало *добру статистичну стійкість, оскільки результати тестів не виходять за межі 0.9–1. Так як запропонований метод є додатковим модулем алгоритму, то аналіз швидкодії прогнозовано показав незначне уповільнення роботи криптосхеми ECIES.*

Ключові слова: циклічний надлишковий код, криптографія, криптостійкість, ECIES, контрольна сума.

A.V. PRYIMAK, O. V. SALIEVA, Y.Y. YAREMCHUK
Vinnytsia National Technical University

INVESTIGATION OF THE POSSIBILITY OF USING OF THE CYCLIC REDUNDANCY CODE ALGORITHM FOR THE INCREASE OF THE ECIES CRYPTOSCHEME STABILITY

The problem of vulnerability of the asymmetric cryptoscheme ECIES to the small subgroups attack is considered, as well as the possibility of using an algorithm of cyclic redundancy code to increase its cryptostability. A method of checking the public key for authenticity is developed that makes it impossible to conduct a successful small subgroups attack, and as a result significantly increases the theoretical stability of this cryptographic algorithm. The essence of the modification is that when generating a public key, the checksum is automatically calculated for it. After exchanging keys, each user generates a check sum of his own public key. This checksum is added to the optional parameter when calculating the message tag using the MAC function. After receiving the message and verifying its tag, it is concluded that all the parameters that participated in its generation are identical. If the checksums do not coincide as a result of a public key substitution, the message tag will also be changed. In this case, it is concluded that the public key of one of the interlocutors has been changed during the exchange and the continued exchange of information with these public keys is dangerous and need to be replaced. The performed statistical testing of the proposed modification of the algorithm showed good statistical stability, since the results of the tests do not exceed 0.9-1, however, generally, both versions showed almost the same results in tests.

Keywords: cyclic redundancy code, cryptography, cryptoscope, ECIES, checksum.

Вступ

Криптографія призначена для передавання захищених даних через незахищену мережу в зашифрованому варіанті, щоб лише один із користувачів, якому призначена ця інформація, міг проаналізувати його. Зв'язок через повідомлення, електронні листи або різні інші режими вимагає високої безпеки. Криптографія еліптичної кривої (ECC) може використовуватися як засіб для шифрування даних, тобто таким чином зберігати конфіденційність вмісту. Серед великої кількості криптосистем, заснованих на ECC, найбільш відома схема інтегрованого шифрування з еліптичною кривою (ECIES), і її можна знайти в декількох криптографічних стандартах.

ECIES (з англ. Elliptic Curve Integrated Encryption Scheme) – це схема шифрування на відкритих ключах, заснована на еліптичних кривих. Ця схема була запропонована Віктором Шоупом в 2001 році. ECIES використовується в різних стандартах, наприклад, ANSI X9.63, IEEE 1363a, ISO 18033-2 та SECG SEC 1 [1, 2].

Безпека ECIES ґрунтується на обчислювальній складності задачі дискретного логарифмування в групі точок еліптичної кривої (ECDLP). Криптографічні алгоритми також можуть гуртуватися на обчислювальної складності завдань факторизації (приклад алгоритму: RSA) і дискретного логарифмування (схема Ель-Гамаля). Однак ECDLP вважається найскладнішою з цих трьох задач, що призводить до важливої переваги ECIES над іншими алгоритмами – невеликий розмір ключа [3].

Важливим недоліком ECIES в порівнянні з іншими криптографічними алгоритмами є існування кількох версій ECIES, описуваних різними стандартами (ANSI X9.63, IEEE 1363a, ISO/IEC 18033-2 і SECG SEC 1). Відмінності між даними стандартами – вибір конкретних функцій і параметрів для реалізації складових ECIES (KA, KDF, ENC, MAC, HASH). Недолік полягає в тому, що неможливо реалізувати версію ECIES, що задовольняє всім стандартам [4].

Попри ряд переваг, даний алгоритм має декілька недоліків [5, 6]:

Однією з відомих можливих атак на дану криптосхему є атака «М'яка вразливість».

Віктор Шоуп довів, що якщо публічний ключ U не включені у вхідні дані функції KDF і, якщо в KDF використовується тільки x-координата розділеного секрету, то ECIES піддається атакам на основі адаптивного шифротексту (Adaptive Chosen Ciphertext Attacks CCA2). Уразливість названа «м'якою», так як жодна з проведених практичних атак не змогла отримати значущу інформацію з використанням цієї

уразливості.

Одне з можливих рішень, запропонованих Шоупом – додати публічний ключ U у вхідні дані функції KDF.

Уразливість при використанні функції XOR.

Шоуп також довів, що схема ECIES може бути вразлива, коли функція XOR використовується при шифруванні повідомлень змінної довжини. Зокрема, це може привести до уразливості до атак на основі адаптивного шифротексту (Adaptive Chosen Ciphertext Attacks CCA2). Можливі рішення:

- зафіксувати довжину відкритого тексту;
- інтерпретувати вихідні дані функції KDF як $k_{ENC} \| k_{MAC}$;
- заборонити використання потокових фільтрів в ECIES (дозволити тільки блокові шифри).

Атака малими підгрупами. Найнебезпечніша відома практично реалізована атака на криптосхему ECIES. Даний тип атак можливий, коли противник спеціально надає невірний публічний ключ. Якщо відправник не перевіряє справжність публічного ключа іншого боку, то противник зможе підмінити публічний ключ на ключ меншого розміру з метою отримання розділеного секрету або отримання інформації про приватний ключ відправника.

У даній роботі розглядається недолік ECIES, пов'язаний з вразливістю до атак малими підгрупами і можливість використання алгоритму циклічного надлишкового коду для знаходження контрольної суми, для перевірки ключів на цілісність, і як результат на справжність, оскільки для проведення успішної атаки малими підгрупами зловмиснику необхідну замінити оригінальний ключ на ключ меншого розміру, що призведе до зміни контрольної суми і у випадку підміни публічного ключа отримувач не буде продовжувати операцію розшифрування.

Постановка задачі та методика дослідження

Провести дослідження криптоалгоритму ECIES щодо можливості використання циклічного надлишкового коду для підвищення його криптостійкості. Запропонувати метод перевірки публічного ключа на справжність за рахунок обчислення контрольної суми. Провести статистичне тестування запропонованого покращення алгоритму та порівняти результати з тестуванням оригінального алгоритму ECIES.

Метод перевірки публічного ключа на справжність

Розглянемо математичний апарат циклічного надлишкового коду. Циклічний надлишковий код (зокрема, CRC8, CRC16, CRC32) застосовується для перевірки цілісності передачі даних. Така контрольна сума проста в реалізації і забезпечує низьку ймовірність виникнення колізій. Циклічні надлишкові коди є частиною стандартів, найпопулярніший і рекомендований IEEE поліном для CRC-32 використовується в Ethernet, FDDI, крім того цей багаточлен є генератором коду Хеммінга [7].

Оцінюючи швидкодію алгоритму CRC-32, можна зробити висновки, що він є значно швидшим за криптографічні хеш-функції. Так, наприклад, для файлу розміром 1 Мб, час знаходження контрольної суми алгоритмом CRC-32 є 0.009 секунди, тоді коли алгоритму SHA-1 необхідно 0.022. Розмір контрольної суми алгоритму CRC-32 є в 5 разів меншим. Оскільки сам публічний ключ не повинен шифруватись і перебуває у відкритому доступі, то достатньо лише зберігати в секреті контрольну суму даного публічного ключа і при передачі даних отримувач буде мати змогу перевірити отриманий публічний ключ на справжність. Тому алгоритмом для обчислення контрольної суми публічного ключа було обрано – CRC-32.

Суть модифікації полягає в тому, що при генерації публічного ключа, для нього автоматично обчислюється контрольна сума C_{SUM}. Після обміну ключами кожен з користувачів генерує контрольну суму публічного ключа свого співрозмовника. Ця контрольна сума додається додатковим параметром при обчисленні тегу повідомлення за допомогою функції MAC. Після отримання повідомлення і перевірки його тегу робиться висновок про ідентичність усіх параметрів, які приймали участь у його генерації. Якщо контрольні суми не збігаються в результаті підміни публічного ключа, то тег повідомлення також буде змінений. В такому випадку робиться висновок, що публічний ключ одного з співрозмовників було змінено при обміні і продовжувати обмін інформацією з цими ж публічними ключами небезпечно і їх потрібно замінити.

Модифікована схема шифрування за алгоритмом ECIES складається з таких етапів:

1. За допомогою методу генерації загального секрету KA особа «А» обчислює загальний секрет $s = p_a * P_B$ (по протоколу Діффі - Хеллмана).

2. Використовуючи отриманий загальний секрет s і метод отримання ключів з ключової і додаткової інформації KDF, особа «А» отримує ключ шифрування k_{ENC}, а також ключ для обчислення імітовставки k_{MAC}.

3. За допомогою симетричного алгоритму шифрування особа «А» шифрує вихідне повідомлення m ключем k_{ENC} і отримує шифротекст c. $c = E(m)$.

4. Особа «А» обчислює контрольну суму публічного ключа особи «Б» за алгоритмом CRC-32. В результаті отримується значення контрольної суми cs_{P_B}.

5. Взявши ключ k_{MAC}, зашифроване повідомлення c і контрольну суму публічного ключа особи «Б», особа «А» обчислює тег повідомлення за допомогою функції MAC. $tag = MAC(c; CS_{P_B})$.

6. Особа «А» відсилає особі «Б» $\{P_A, tag, c\}$ $\{P_A, tag, c\}$.

Процес дешифрування залишається без змін, за виключенням того, що одержувачу буде необхідно перевірити тег повідомлення разом з контрольною сумою.

Таким чином при можливій підміні публічного ключа зловмисником, на менший за розміром, його контрольна сума буде змінюватись і як результат буде змінюватись тег повідомлення.

Графічне представлення схеми роботи запропонованої модифікації зображено на рисунку 1.

Для порівняння швидкості шифрування повідомлення оригінальним алгоритмом ECIES і алгоритмом ECIES із вбудованим методом перевірки публічного ключа на справжність, за допомогою використання циклічного надлишкового коду, було обрано три різні ключі, довжинами 256, 384, 512 біт, а також повідомлення довжиною 10000 байт. Результати порівняння представлені в таблиці 1.

Виходячи з результатів наведених в таблиці 1, можна зробити висновок, що модифікований ECIES дещо сповільнює роботу алгоритму при шифруванні, оскільки саме на етапі шифрування генерується контрольна сума публічного ключа співрозмовника та додається до тегу повідомлення. Також варто зазначити, що результати швидкодії дешифрування оригінального і модифікованого алгоритмів ECIES суттєво не відрізняються. Різниця складає лише 2–4 мс.

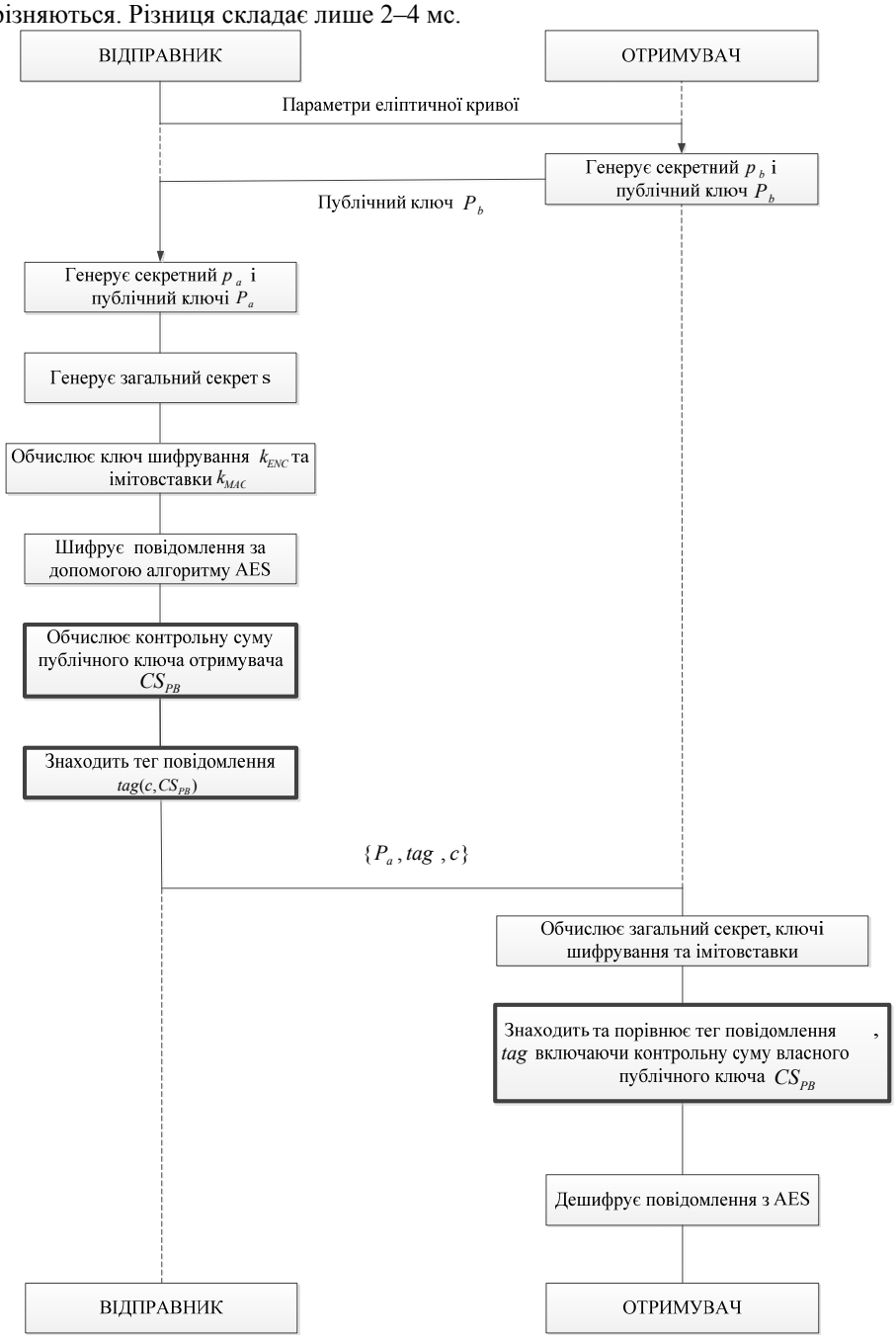

Рис. 1. Схема роботи модифікованого методу шифрування ECIES

Таблиця 1

Порівняння швидкості шифрування та дешифрування оригінального алгоритму та модифікованого

Довжина ключа (біт)	Час (мс)			
	Шифрування		Дешифрування	
	Оригінальний ECIES	Модифікований ECIES	Оригінальний ECIES	Модифікований ECIES
256	81	88	61	63
384	69	76	50	54
512	135	141	109	112

Графічні результати проведених статистичних замірів швидкодії оригінального алгоритму ECIES та модифікованого при використанні трьох різних за довжиною ключів для одного ж повідомлення розміром 10000 байт представлено на рисунку 2.

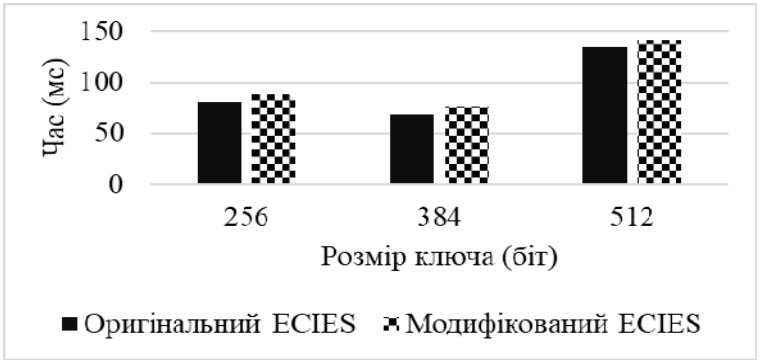

Рис. 2. Результати статистичних замірів швидкості шифрування оригінального алгоритму ECIES та модифікованого

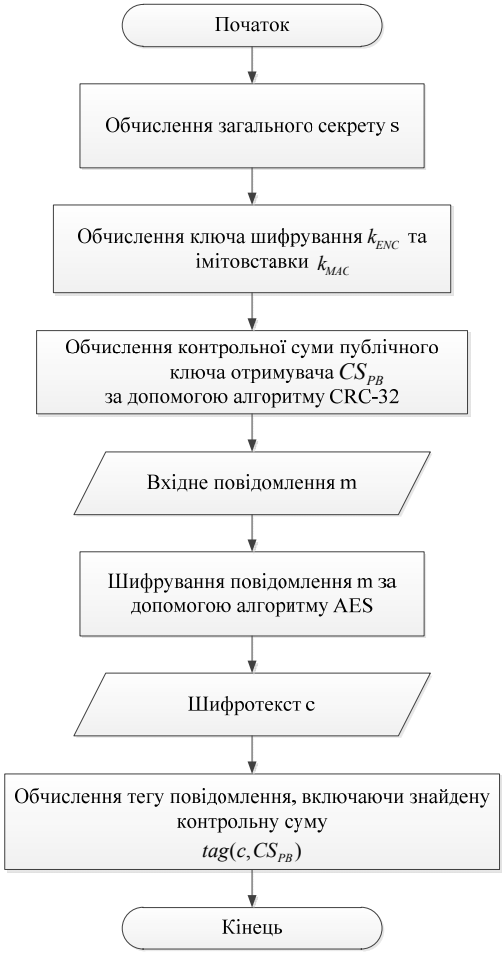

Рис. 3. Схема роботи алгоритму програми на основі запропонованого методу

Виходячи з результатів порівняння швидкості шифрування оригінальним алгоритмом ECIES та

модифікованим, можна зробити висновок, що запропонований метод перевірки публічних ключів на справжність дещо сповільнює роботу алгоритму.

У випадку реалізації роботи процесу шифрування ECIES, з використанням запропонованого методу перевірки публічного ключа на справжність за рахунок циклічного надлишкового коду на програмному рівні, блок-схема такого додатку буде мати такий вигляд (рис. 3):

Крім порівняння швидкості шифрування повідомлення оригінальним алгоритмом ECIES і алгоритмом ECIES із вбудованим запропонованим методом перевірки публічного ключа на справжність, доцільно також порівняти загальне навантаження на роботу процесора комп'ютера на якому відбувається той чи інший процес алгоритму.

Тестування проведено на двох операційних системах – Windows 10 та Linux/Ubuntu 18.02, з використанням процесора Intel Core i5 – 5350H, 2.4 ГГц. Результати порівняння наведені в таблиці 2.

Таблиця 2

Порівняння параметрів навантаження на процесор оригінального алгоритму та модифікованого ECIES

Розмір повідомлення для шифрування (байт)	Навантаження на процесор у Windows		Навантаження на процесор у Linux	
	Оригінальний ECIES	Модифікований ECIES	Оригінальний ECIES	Модифікований ECIES
256	0.26%	0.28%	0.16%	0.16%
384	0.31%	0.33%	0.18%	0.18%
512	0.36%	0.39%	0.23%	0.23%

З результатів порівняння, наведених вище, можна зробити висновок, що запропонований метод перевірки публічного ключа на справжність, не суттєво впливає на навантаження процесора при виконанні процесу шифрування і дешифрування. Отримана різниця в навантаженні знаходиться в межах 0.02–0.03% і так само як і в порівняльному аналізі швидкодії алгоритмів простежується залежність, що при зміні довжини ключів, різниця в навантажені майже не змінюється, тобто ця різниця є стала.

Для дослідження статистичної безпеки асиметричного алгоритму ECIES з вбудованим запропонованим методом перевірки публічного ключа на справжність за рахунок циклічного надлишкового коду буде використано пакет статистичних тестів NIST STS (National Institute of Standard and Technologies Statistical TestSuite). До його складу входять 15 статистичних тестів, метою яких є визначення міри випадковості двійкових послідовностей, породжених або апаратними, або програмними генераторами випадкових чисел. Ці тести побудовані на різних статистичних властивостях, притаманних тільки випадковим послідовностям [8].

Основними параметрами для проходження тестів було обрано:

– довжина ключа –512 біт;

– кількість тестів – 188.

В таблиці 3 представлена порівняльна характеристика результатів проходження усіх 15-и тестів оригінального алгоритму ECIES та з вбудованим методом перевірки публічного ключа на справжність.

Таблиця 3

Відсотки проходження кожного з 15 тестів для ключа довжиною 512 біт

Назва статистичного тесту	**Оригінальний ECIES**	**Модифікований ECIES**
Частотний (монобітний) тест	99%	97%
Частотний тест всередині блоку	97%	98%
Послідовний тест	98%	100%
Перевірка максимальної довжини серії в блоці	98%	96%
Перевірка рангу двійкової матриці	98%	98%
Спектральний тест на основі дискретного перетворення Фур'є	96%	99%
Перевірка шаблонів, які не перекриваються	98%	95%
Перевірка шаблонів, які перекриваються	98%	97%
Універсальний тест Маурера	95%	98%
Перевірка лінійної складності	98%	99%
Перевірка серій	98%	98%
Ентропійний тест	96%	98%
Перевірка накоплених сум	96%	93%
Перевірка випадкових відхилень	96%	99%
Перевірка випадкових відхилень (модифікація)	95%	98%

Як видно з результатів наведених у таблиці 3, оригінальний алгоритм показує дещо гірші показники порівняно з запропонованою його модифікацією.

Так, наприклад, частотний тест всередині блоку, послідовний тест, спектральний тест на основі дискретного перетворення Фур'є, універсальний тест Маурера, тест на перевірку лінійної складності, ентропійний тест, тест на перевірку випадкових відхилень та тест на перевірку випадкових відхилень (модифікація), показали, що модифікований алгоритм ECIES з вбудованим запропонованим методом перевірки публічного ключа на справжність має вищі показники на 1–3%.

Для наглядності доцільно також провести порівняння статистичних портретів модифікованого та оригінального алгоритму ECIES.

З рисунку 4 видно, що результати тестів модифікованого алгоритму ECIES не виходять за межі 0.9 - 1, що показує високу статистичну надійність даного методу. Даний результат свідчить про добру статистичну стійкість модифікованого алгоритму.

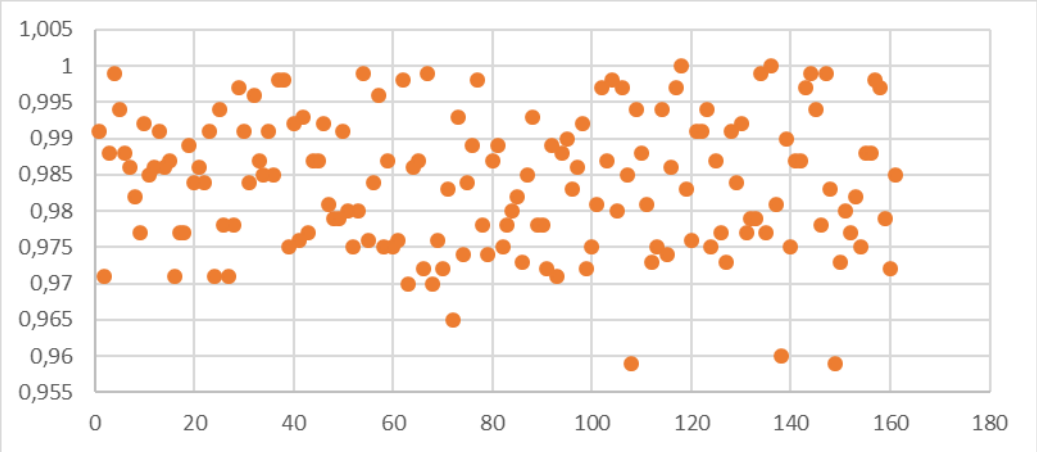

Рис. 4. Результати тестування модифікованого алгоритму ECIES

На відміну від результатів тестування модифікованого алгоритму, результати оригінального алгоритму ECIES (рис. 5) знаходяться в більш широкому діапазоні.

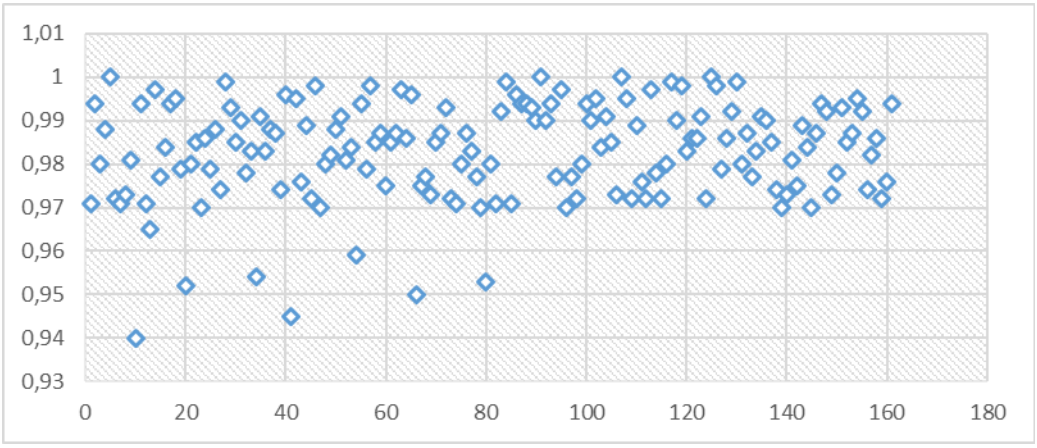

Рис. 5. Результати тестування оригінального алгоритму ECIES

На рисунку 6 представлено порівняння результатів тестування для кожного тесту з статистичного пакету NIST оригінального алгоритму і запропонованої модифікації.

Як видно з графіку на рисунку 5, алгоритм із вбудованим запропонованим методом перевірки публічного ключа на справжність показав кращі результати у більшості тестах.

Виходячи з наведених вище результатів статистичних тестів, можна зробити висновок, що модифікований алгоритм ECIES з вбудованим запропонованим методом перевірки публічного ключа на справжність за рахунок використання циклічного надлишкового коду має добру статистичну стійкість, оскільки він показав вищі результати на восьми тестах з п'ятнадцяти, а результати тестів ECIES не виходять за межі 0.9–1.

Висновки

Проведено дослідження криптоалгоритму ECIES та можливість використання математичного апарату циклічного надлишкового коду для підвищення його криптостійкості. Було розроблено метод перевірки публічного ключа на справжність, що унеможливлює проведення успішної атаки малими підгрупами і як результат значно підвищує теоретичну стійкість даного криптоалгоритму.

Проведено статистичне тестування запропонованої модифікації алгоритму та порівняно результати з тестуванням оригінального алгоритму ECIES. Результати статистичного тестування показали високу статистичну надійність запропонованого методу. Також варто зазначити, що модифікований алгоритм

ECIES з вбудованим запропонованим методом перевірки публічного ключа на справжність показав вищі показники на 1–3% е вісьми з п’ятнадцяти тестів.

Крім того було проведеного аналіз швидкодії, який показав, що запропонована модифікація дещо уповільнює роботу алгоритму, в середньому на 2–7 мс для повідомлення розміром 10000 байт.

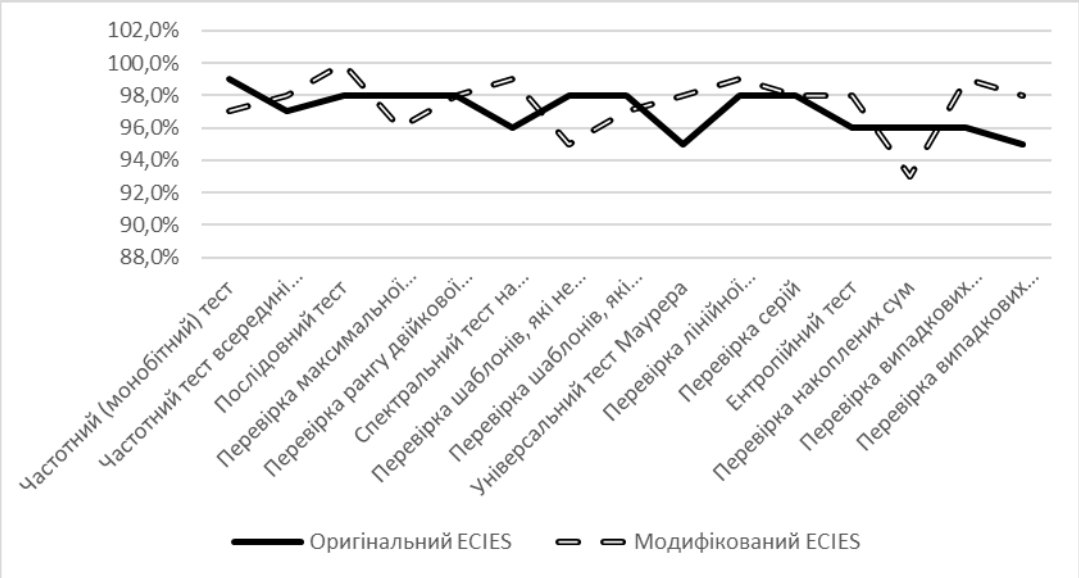

Рис. 6. Порівняння результатів тестування

Література

1. Setiadi I. Elliptic curve cryptography: Algorithms and implementation analysis over coordinate systems / I. Setiadi, A. Kistijantoro, A. Miyaji // 2015 2nd International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), Chonburi. – 2015. – С. 1–6.

2. Gayoso Martínez V. A Survey of the Elliptic Curve Integrated Encryption Scheme / V. Gayoso Martínez, L. Hernández Encinas, C. Sánchez Ávila // JOURNAL OF COMPUTER SCIENCE AND ENGINEERING. – 2010. – №2. – С. 7–13.

3. Ms. Manali Dubal. Achieving Authentication and Integrity using Elliptic Curve Cryptography Architecture / Ms. Manali Dubal, Ms. Aaradhana Deshmukh // International Journal of Computer Applications (0975 – 8887). – 2014. – №24. – С. 11–15.

4. A comparison of the standardized versions of ECIES / V. Gayoso Martínez, F. Hernández, Á. lvarez, L. Hernández Encinas, C. Sánchez Á vila // 2010 Sixth International Conference on Information Assurance and Security, Atlanta, GA. – 2010. – С. 1–4.

5. Koffka K. The Security of Elliptic Curve Cryptosystems - A Survey / Khan Koffka // Global Journal of Computer Science and Technology. – 2015. – №5. – С. 24–35.

6. Integrated Encryption Scheme // Wikipedia. – 2016. URL: https://en.wikipedia.org/wiki/Integrated_Encryption_Scheme.

7. Krishnaveni V. Analysis of Efficient CRC Implementation Configurations / V. Krishnaveni, S. V.V.N., N.J. Lakshmi // International Journal of Engineering In Advanced Research Science and Technology ISSN: 2278-256. – 2016. – №4. – С. 7880–7888.

8. Pareschi F. On Statistical Tests for Randomness Included in the NIST SP800-22 Test Suite and Based on the Binomial Distribution / F. Pareschi, R. Rovatti, G. Setti // IEEE Transactions on Information Forensics and Security. – 2012. – №2. – С. 491–505.

Рецензія/Peer review : 27.1.2019 р. Надрукована/Printed :16.2.2019 р.
Рецензент: