УДК 004.056.55

А. В. Приймак, Ю. Є. Яремчук
Вінницький національний технічний університет
Хмельницьке шосе, 95, 21021 Вінниця, Україна

Підвищення стійкості криптоалгоритму RSA за рахунок генетичної оптимізації вихідного повідомлення

Розглянуто існуючі асиметричні алгоритми шифрування інформації. Описано їхні переваги та недоліки. Проведено дослідження алгоритму RSA щодо можливості підвищення його криптостійкості. Запропоновано метод оптимізації вихідного повідомлення за допомогою генетичного алгоритму. Представлено статистичне тестування запропонованого покращення алгоритму, яке показало, що отримані показники статистичної безпеки є вищими за показники оригінального алгоритму на 1–3 %.

Ключові слова: *криптостійкість, RSA, детермінованість, генетичний алгоритм.*

Вступ

У зв'язку з великою кількістю даних, які кожен день передаються через комп'ютерні мережі та зберігаються у хмарних середовищах, мережева безпека стала одним із найважливіших аспектів роботи в мережі. Для захисту інформації у мережі існує багато підходів, серед яких одним з найефективніших і найпопулярніших є криптографія. За допомогою криптографії вирішується питання забезпечення конфіденційності, цілісності й автентичності інформації (захищеного передавання даних, обміну інформацією чи її збереження). За допомогою шифрування інформації вирішується основна задача криптографії — забезпечення конфіденційності, в першу чергу, з метою захисту її від несанкціонованого доступу. Суть шифрування полягає в тому, що для проведення оберненої операції необхідно знати секретний ключ, в іншому випадку отримати вихідне повідомлення буде неможливо [1].

Сучасна криптографія поділяється на симетричну і асиметричну. До переваг асиметричної криптографії відносять легкість управління ключами у великій мережі, а також легкість обміну ключами, оскільки не потрібен секретний канал для передавання кожного ключа обом сторонам [2]. Також варто зазначити, що алгоритми асиметричної криптографії у багатьох країнах є основою національних стан-

дартів шифрування, широко використовуються в інтернет-протоколах обміну даних, у криптовалюті та інших популярних онлайн-сервісах і ресурсах. На даний момент найбільш відомими та найбільш поширеними криптографічними асиметричними алгоритмами є: ElGamal, DSA, ECDSA та RSA [3].

Для порівняння вищеперерахованих алгоритмів доцільно провести аналіз кожного з них, описати їхні переваги та недоліки.

1. Схема Ель-Гамаля (Elgamal) — асиметричний алгоритм шифрування, що побудований на труднощі обчислення дискретних логарифмів у кінцевому полі. Криптосистема включає в себе алгоритм шифрування і алгоритм цифрового підпису. Схема Ель-Гамаля лежить в основі колишніх стандартів електронного цифрового підпису в США (DSA) [4].

2. DSA — алгоритм цифрового підпису — розроблений NSA як частина стандарту цифрового підпису (DSS). Використовується для цифрового підпису. Не виконує шифрування, але існують реалізації, які можуть виконувати шифрування за допомогою шифрування RSA або ElGamal). DSA генерує цифровий підпис, що складається з двох 160-бітових послідовностей безпосередньо з приватного ключа та хеш-даних, які підлягають підписанню. Відповідний відкритий ключ можна використовувати для перевірки підпису. Процес перевірки є занадто повільним. Запатентований, але доступний безкоштовно. Безпека DSA базується на проблемі обчислення дискретного логарифму [5].

3. ECDSA (Elliptic Curve Digital Signature Algorithm) — алгоритм з відкритим ключем для створення цифрового підпису, аналогічний за своєю будовою DSA, але визначений, на відміну від нього, не над кільцем цілих чисел, а в групі точок еліптичної кривої. ECC базується на теорії еліптичних кривих і вирішує проблему «Еліптичної кривої дискретного завдання логарифму (ECDLP)», яку дуже важко вирішити. Ключі ECC сильніші, ніж ключі RSA та DSA, оскільки алгоритм важче зламати. Як і в DSA, він вимагає хорошого джерела випадкових чисел. Якщо джерело не є хорошим, то може бути витік приватного ключа. Хоча ECDLP важко зламати, існує багато атак, які можуть успішно дешифрувати ECC, якщо крива, що обрана при реалізації, є поганою. Для високої криптостійкості ECC необхідно використовувати SafeCurves, наприклад Curve25519 [6].

4. RSA побудований на складності задачі факторизації великих чисел. Користувач RSA створює, а потім публікує відкритий ключ на основі двох великих простих чисел разом із допоміжним значенням. Прості числа повинні зберігатися в таємниці. Будь-хто може використовувати відкритий ключ для шифрування повідомлення, але за допомогою опублікованих методів, і якщо відкритий ключ достатньо великий, лише кожен, хто знає про прості числа, може розшифровувати це повідомлення. Основними недоліками є детермінованість шифротексту та вразливість до атаки на основі підібраного шифротексту. Варто зазначити, що RSA підтримується всіма версіями SSL/TLS, протоколами, які регулюють безпечний обмін даними в мережі Інтернет [7].

У табл. 1 представлено результати порівняння найбільш популярних асиметричних алгоритмів шифрування на основі їхньої криптостійкості.

Виходячи з аналізу асиметричних алгоритмів шифрування, можна зробити висновок, що алгоритм RSA є найбільш поширеним та одним з найбільш застосовуваних на сьогоднішній день, він використовується різними стандартами для за-

безпечення захисту інформації. При всіх його перевагах, цей алгоритм має два суттєві недоліки, а отже залишається актуальним вирішення проблеми детермінованості шифротексту і, як результат, підвищення криптостійкості даного алгоритму. У цьому зв'язку пропонується звернути увагу на математичний апарат генетичних алгоритмів, який має потенційні можливості вирішення цієї проблеми.

Таблиця 1. Порівняння асиметричних алгоритмів шифрування

Алгоритм	**Ключ**	**Призначення**	**Криптостійкість, MIPS**
RSA	До 4096 біт	Шифрування та підпис	$3*10^{20}$ для ключа 2048 біт. Швидкість роботи відносно невисока.
ElGamal	До 4096 біт	Шифрування та підпис	За однакової довжини ключа криптостійкість така ж як і в RSA.
DSA	До 1024 біт	Тільки цифровий підпис	Криптостійкість і швидкість роботи вище ніж у ECDSA.
ECDSA	До 4096 біт	Шифрування та підпис	Криптостійкість і швидкість роботи вище ніж у ElGamal.

Постановка задачі та методика дослідження

Провести дослідження криптоалгоритму RSA щодо можливості підвищення його стійкості за допомогою генетичного алгоритму. Запропонувати метод оптимізації вихідного повідомлення за допомогою генетичного алгоритму. Провести статистичне тестування запропонованого покращення алгоритму та порівняти результати з тестуванням оригінального алгоритму RSA.

Метод оптимізації вихідного повідомлення генетичним алгоритмом

Розглянемо математичний апарат генетичних алгоритмів, які базуються на концепції «виживання найбільш придатних» і працюють над тим, щоб знайти оптимальне або майже оптимальне рішення для оптимізації завдань. Генетичний алгоритм має за мету вирішення завдань шляхом моделювання спрощеної версії генетичних процесів [8].

Використовуючи три основні властивості генетичного алгоритму (оператор відбору, схрещення та мутація), можна рандомізувати вихідне повідомлення, яке в результаті шифрування алгоритмом RSA перетвориться на стохастичний шифротекст, який уже не буде детермінованим і слабким до атаки на основі підібраного шифротексту, і при цьому підвищити криптостійкість даного алгоритму.

Пропонується метод оптимізації вихідного повідомлення генетичним алгоритмом, який складається з 8 наступних кроків.

1. Конвертується вихідне повідомлення m, звичайний символьний текст, в двійковий код. Як результат отримується двійкова послідовність M.

2. Розбивається конвертоване повідомлення M на 2 рівні частини M_1 та M_2. Якщо M_1 та M_2 нерівні, то дописуються нулі.

3. Генерується випадкова точка схрещення S (від 0 до кількості біт у M_1 та M_2).

4. Відбувається схрещення M_1 та M_2 за точкою S. В результаті отримуються M_1' та M_2'.

5. Об'єднуються M_1' та M_2' в одну послідовність біт для подальшого проведення процесу мутації — M'.

6. Генеруються 3 випадкових числа в межах від 0 до кількості біт у M'. Ці числа є номерами бітів, які будуть замінені під час мутації. Всі 3 числа будуть зберігатись як одна послідовність R.

7. Проводиться операція мутації за вибраними числами з попереднього кроку. Результуюча послідовність M''.

8. Формується вихідне повідомлення m' після процесу оптимізації генетичним алгоритмом, яке буде приймати подальшу участь у процесі шифрування алгоритмом RSA. Оптимізоване повідомлення матиме такий вигляд — [M'', S, R].

Варто зазначити, що шифротекст, що отриманий з оптимізованого вихідного повідомлення, буде завжди унікальним, навіть для одного й того ж повідомлення. Тому запропонований метод вирішує проблему з можливою атакою на основі підібраного шифротексту, оскільки порівняння шифротексту немає сенсу, так як послідовність, яка шифрується, не матиме жодної прямої залежності з вихідним повідомленням.

Графічне представлення запропонованого методу оптимізації вихідного повідомлення генетичним алгоритмом представлено на рис. 1.

Оскільки метод оптимізації вихідного повідомлення генетичним алгоритмом є додатковим модулем роботи всього алгоритму шифрування, то необхідно враховувати також швидкодію роботи самого алгоритму.

Для порівняння швидкості шифрування та дешифрування повідомлення оригінальним алгоритмом RSA і алгоритмом RSA із вбудованим запропонованим методом оптимізації вихідного повідомлення генетичним алгоритмом, було обрано ключ довжиною 1024 біт і 4 різні повідомлення довжинами 700, 1000, 1500 та 2000 байт відповідно. Результати порівняння наведено в табл. 2.

Таблиця 2. Порівняння швидкості шифрування та дешифрування

Розмір повідомлення (байт)	Час (мс)			
	Шифрування		Дешифрування	
	Оригінальний RSA	Модифікований RSA	Оригінальний RSA	Модифікований RSA
700	19	23	2134	2168
1000	23	25	2874	2901
1500	36	37	4359	4372
2000	47	47	5772	5781

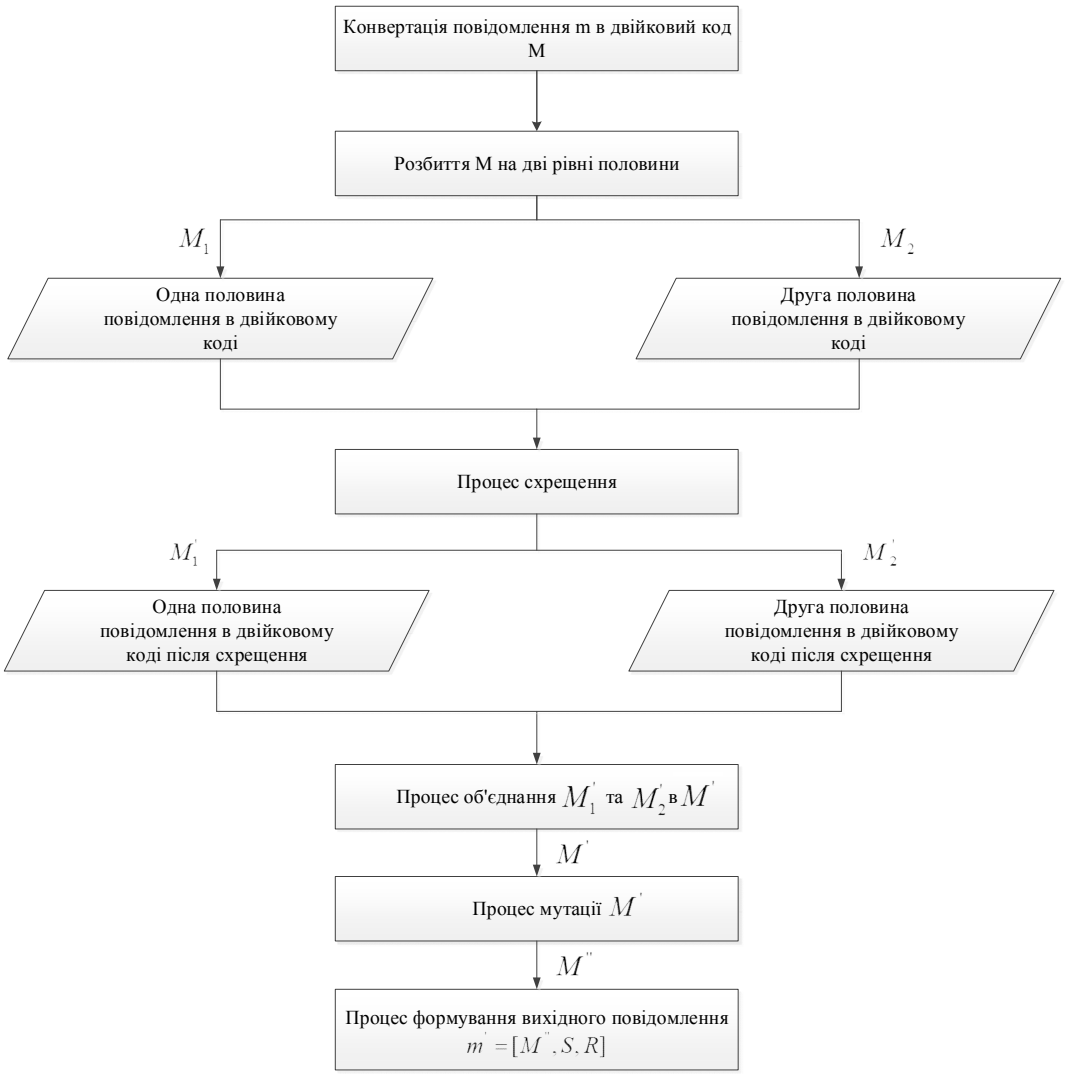

Рис. 1. Схема роботи методу оптимізації вихідного повідомлення

Графічні результати проведених статистичних замірів швидкодії оригінального алгоритму RSA та модифікованого при шифруванні чотирьох різних за розміром повідомлень представлено на рис. 2.

Виходячи з результатів порівняння швидкості шифрування оригінальним алгоритмом RSA та модифікованим, можна зробити висновок, що запропонований метод оптимізації вихідного повідомлення генетичним алгоритмом майже не сповільнює роботу алгоритму.

Для дослідження статистичної безпеки асиметричного алгоритму RSA із вбудованим запропонованим методом оптимізації вихідного повідомлення буде використано пакет статистичних тестів NIST STS. 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, що притаманні тільки випадковим послідовностям.

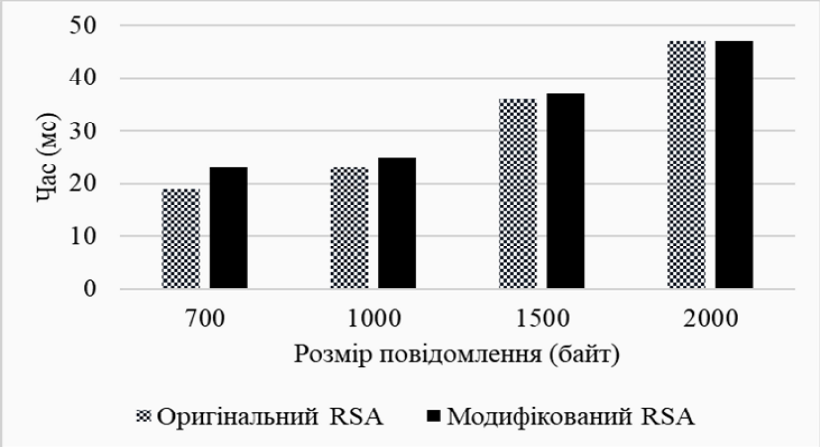

Рис. 2. Результати статистичних замірів швидкості шифрування оригінального алгоритму RSA та модифікованого

Основними параметрами для проходження тестів було обрано:

1) довжину ключа — 1024 біт;

2) кількість тестів — 188.

У табл. 3 представлено порівняльну характеристику результатів проходження усіх 15-ти тестів оригінального алгоритму RSA та із вбудованим методом оптимізації вихідного повідомлення.

Таблиця 3. Відсотки проходження кожного з 15 тестів для ключа довжиною 1024 біт

Назва статистичного тесту	Оригінальний RSA	Модифікований RSA
Частотний (монобітний) тест	99 %	99 %
Частотний тест всередині блоку	97 %	98 %
Послідовний тест	98 %	100 %
Перевірка максимальної довжини серії у блоці	98 %	98 %
Перевірка рангу двійкової матриці	98 %	99 %
Спектральний тест на основі дискретного перетворення Фур'є	96 %	99 %
Перевірка шаблонів, які не перекриваються	98 %	99 %
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, які наведено в табл. 3, оригінальний алгоритм показує гірші показники порівняно із запропонованою його модифікацією. Десять з п'ятнадцяти тестів показали, що модифікований алгоритм RSA із вбудованим запропонованим методом оптимізації вихідного повідомлення має вищі показники на 1–3 %.

Гірші показники модифікований алгоритм RSA мав лише по проходженню тесту на перевірку шаблонів, які перекриваються та тесту на перевірку накопичених сум.

Виходячи з наведених вище результатів статистичних тестів, можна зробити висновок, що модифікований алгоритм RSA з вбудованим запропонованим методом оптимізації вихідного повідомлення має вищий рівень статистичної безпеки, ніж оригінальний алгоритм RSA, оскільки він показав вищі результати на десятьох тестах з п'ятнадцяти.

Для наочності доцільно також провести порівняння статистичних портретів модифікованого та оригінального алгоритму RSA.

З рис. 3 видно, що результати тестів модифікованого алгоритму RSA не виходять за межі 0,9–1, що показує високу статистичну надійність даного методу.

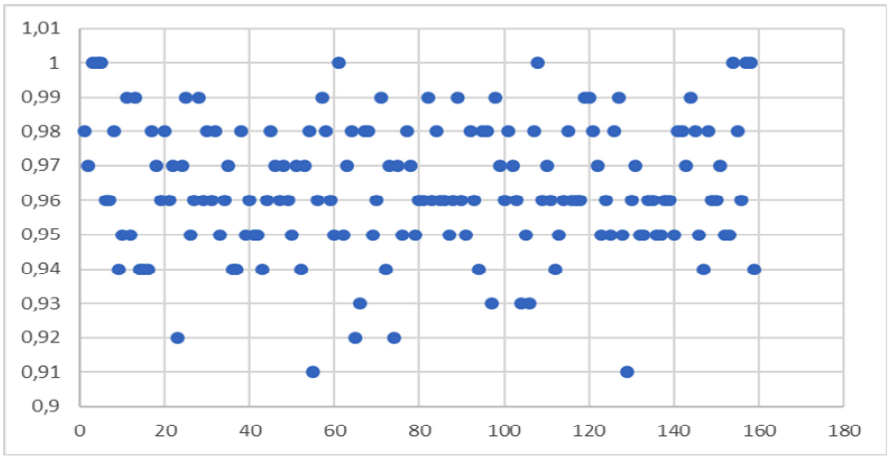

Рис. 3. Результати тестування модифікованого алгоритму RSA

На відміну від результатів тестування модифікованого алгоритму, результати оригінального алгоритму RSA (рис. 4) знаходяться в більш широкому діапазоні.

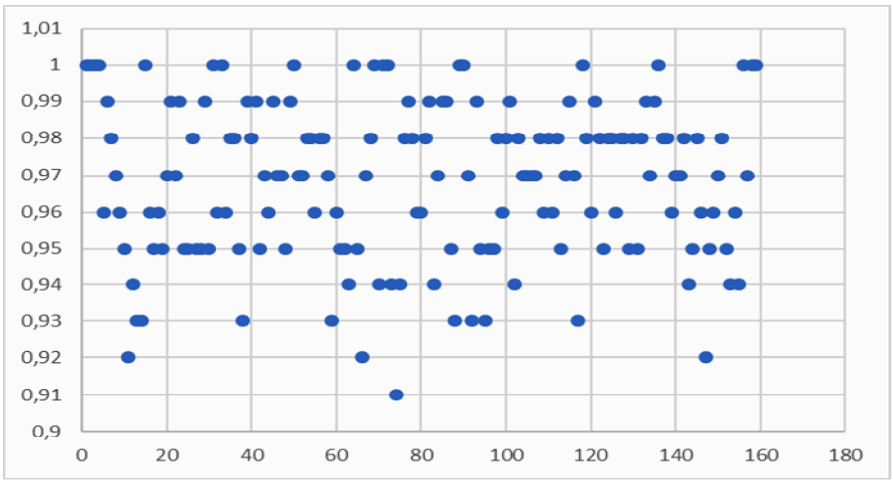

Рис. 4. Результати тестування оригінального алгоритму RSA

На рис. 5 представлено порівняння результатів тестування для кожного тесту зі статистичного пакету NIST оригінального алгоритму та запропонованої модифікації.

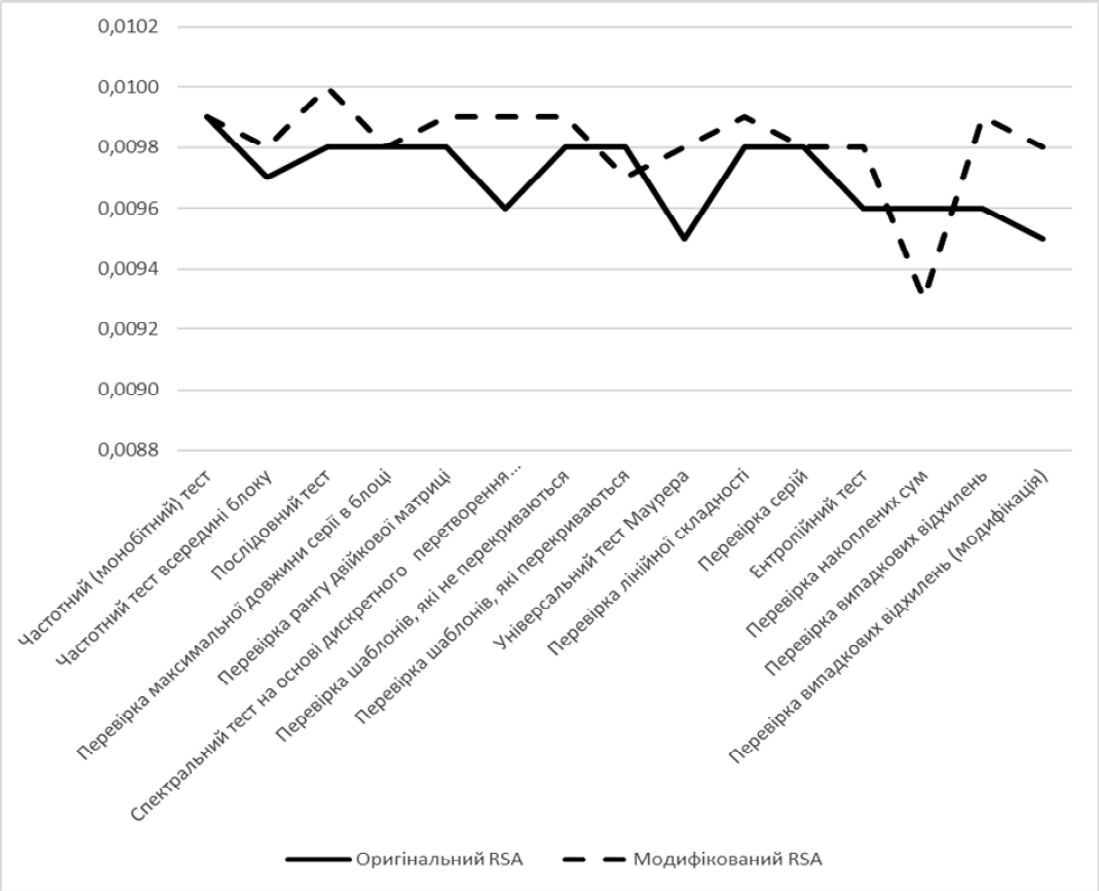

Рис. 5. Графічне порівняння результатів тестування

Як видно з графіка на рис. 5, алгоритм із вбудованим запропонованим методом оптимізації вихідного повідомлення показав кращі результати майже в усіх тестах, що свідчить про його вищий рівень статистичної безпеки.

Висновки

Проведено експериментальне дослідження алгоритму RSA та можливість використання генетичного алгоритму для підвищення його криптостійкості.

На основі трьох основних властивостей генетичного алгоритму було розроблено метод оптимізації вихідного повідомлення для вирішення проблеми детермінованості шифротексту та унеможливлення проведення успішної атаки на основі підібраного шифротексту. Запропонований метод складається з восьми кроків.

Проведено статистичне тестування запропонованого покращення алгоритму та порівняно результати з тестуванням оригінального алгоритму RSA. Результати статистичного тестування показали, що модифікований алгоритм RSA із вбудованим запропонованим методом оптимізації вихідного повідомлення має вищі показ-

ники на 1–3 % в десяти з п’ятнадцяти тестів, що свідчить про вищий рівень статистичної безпеки.

1. Jana Bappaditya, Chakraborty Moumita, Tamoghna Mandal, Kule Malay. An Overview on Security Issues in Modern Cryptographic Techniques. Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT). 2018.

2. Charles Joseph, Carol I., Mahalakshmi S. Big Data Security an Overview. *International Research Journal of Engineering and Technology (IRJET)*. 2018. Issue 2. Vol. 5. P. 130–134. URL: https://www.irjet.net/ archives/V5/i2/IRJET-V5I232.pdf

3. R. Sivakumar, B. Balakumar, V. Arivu Pandeeswaran. A Study of Encryption Algorithms (DES, 3DES and AES) for Information Security. *International Research Journal of Engineering and Technology (IRJET)*. 2018. Issue 4. Vol. 5. P. 4133–4137.

4. Joye M. Secure ElGamal-Type Cryptosystems Without Message Encoding. The New Codebreakers. *Lecture Notes in Computer Science*. 2016. P. 470–478.

5. Anu, Shree Divya, Sindhu Rashmi. Analysis of Cryptography and Comparison of its Various Techniques. *International Journal of Advanced Research in Computer Science*. 2017. Issue 5. Vol. 8. P. 688–691.

6. Rabah Kefa. Implementing Elliptic Curve Digital Signature Algorithm (ECDSA) Schemes. *Mara Research Journal of Computer Science & Security*. 2017. Vol. 1. P. 29–50. ISSN 2518-8453.

7. Priyadarshini Patila, Prashant Narayankarb, Narayan D.G., Meena S.M. A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish. Elsevier. 2016. P. 617–624.

8. Приймак А., Яремчук Ю. Підвищення стійкості шифру BLOWFISH на основі оптимізації слабких ключів генетичним алгоритмом. *Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні*. 2018. Вип. 1(35). С. 106–115.

Надійшла до редакції 10.12.2018