УДК 004.94:656.1 DOI: https://doi.org/10.33577/2312-4458.20.2019.20-28

Т.А. Матвейчук, В.Д. Смичок

Національна академія сухопутних військ імені гетьмана Петра Сагайдачного, Львів

ДОСЛІДЖЕННЯ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ АЛГОРИТМІВ АСИМЕТРИЧНОГО ШИФРУВАННЯ

У роботі за допомогою розробленого програмного продукту було ґрунтовно досліджено та проведено порівняльний аналіз асиметричних алгоритмів шифрування текстових даних, їх переваг та недоліків, криптографічної стійкості, дано експериментальну оцінку їх характеристик стосовно ефективності використання ними пам'яті комп'ютера, тривалості процесів генерування ключів, шифрування та дешифрування, пропускної здатності алгоритмів, розмірностей ключів, об'ємів зашифрованих і дешифрованих файлів. На підставі отриманих результатів надано рекомендації щодо застосування розглянутих методів шифрування. В якості асиметричних алгоритмів шифрування обрано алгоритми RSA та ElGamal.

Ключові слова: *асиметричні алгоритми, шифрування, дешифрування, криптостійкість, RSA, ElGamal.*

Постановка проблеми

В сучасному інформаційному суспільстві велика кількість послуг забезпечується за допомогою комп'ютерних мереж та інформаційних технологій, невпинний розвиток яких надзвичайно загострює питання інформаційної безпеки. Інформація, що представлена в цифровому вигляді, має бути надійно захищена від багатьох загроз: несанкціонованого доступу, підробки, витоку інформації, розголошення таємної інформації і таке інше. Тому особливої актуальності сьогодні набуває проблематика ефективних методів захисту інформації у інформаційних системах.

Загрози безпеці інформації пов'язані з тим, що, з одного боку, розширилось використання комп'ютерних мереж, по яких передаються великі потоки інформації, доступ до якої стороннім особам суворо заборонений. З іншого боку, поява сучасних потужних комп'ютерів, розвиток інформаційних технологій і нейронних обчислень зробили можливою дискредитацію криптографічних систем, які до цього вважалися стійкими до криптоаналізу.

Особливого значення інформаційна безпека набула у військовій сфері. Найбільш уразливим елементом в структурі Збройних сил України з точки зору інформаційного протиборства є автоматизовані системи військового призначення, тобто всі обчислювальні системи, системи управління та зв'язку. Активізація боротьби в кіберпросторі стає невід'ємною складовою військових конфліктів [1].

Аналіз останніх досліджень і публікацій

Гострота та актуальність інформаційної безпеки в системі національної безпеки обумовила інтерес вітчизняних та зарубіжних вчених до даної проблеми, про це свідчать наукові праці В.О. Ананьїна, В.М. Богуша, Т.А. Коркішко, В.М. Литвиненка, В.Ф. Ємця, А.О. Мельника, В.Б. Толубка, О.К. Юдіна та ін.

Головну роль у будь-якій системі безпеки відіграють криптографічні методи, пов'язані з алгоритмами шифрування даних. Ці алгоритми витрачають значну кількість часу і ресурсів системи.

Найпоширенішим алгоритмом асиметричного шифрування є алгоритм RSA. Він був запропонований трьома науковцями – Р. Рівестом (R. Rivest), А. Шаміром (A. Shamir) і Л. Адльманом (L. Adleman) в 1977-1978 роках. У 1993 році метод RSA був оприлюднений і прийнятий як стандарт (PKCS # 1: RSAEncryptionstandart).

Безпека алгоритму RSA ґрунтується на складності розкладання на множники великих чисел, а саме – на винятковій складності задачі визначення секретного ключа на підставі відкритого ключа, оскільки для цього потрібно буде вирішити задачу про існування дільників цілого числа. Найбільш криптостійкі системи використовують 1024-бітові і більші числа.

В якості другого несиметричного алгоритму було обрано алгоритм шифрування за схемою ElGamal. Цей алгоритм існує вже тривалий час (був запропонований Тахером Ель-Гамалем в 1984 році), він став першим повноцінним алгоритмом із відкритим ключем, який можна використовувати для шифрування і цифрових підписів, що не захищений патентами в США та світі (патент на алгоритм Діффі-Хеллмана закінчився у 1997 році). Крім того, він є відносно простим для розуміння та

реалізації. Алгоритм ElGamal також є достатньо популярним. Вже багато років він протистоїть інтенсивному криптоаналізу.

Безпека алгоритму ElGamal ґрунтується на складності обчислення дискретного логарифму у скінченному полі. Якщо підносити число до ступеня в скінченному полі достатньо легко, то відновити аргумент за значенням (тобто знайти логарифм) є доволі складною задачею. Відкритий і закритий ключі шифрування є функціями двох великих (1024 - 2048 розрядів в двійковому представленні або навіть більше) простих чисел.

Системи RSA і ElGamal достатньо ґрунтовно описані в багатьох наукових джерелах [2–4].

Формулювання мети статті

Задачею експериментального дослідження є проведення різноманітних експериментів з метою порівняння алгоритмів асиметричного шифрування текстових даних стосовно використання ними пам'яті, тривалості процесів генерування ключів, шифрування та дешифрування, пропускної здатності алгоритмів, розмірів файлів і ключів. У якості асиметричних алгоритмів вибрано алгоритми шифрування RSA та ElGamal.

Основний матеріал

Порівняння криптостійкості алгоритмів. З точки зору практичної реалізації, як програмним, так і апаратним способом відчутної різниці між алгоритмами RSA та ElGamal немає, однак у криптостійкості вони помітно відрізняються.

Якщо розглядати задачу розкладання довільного цілого числа довжиною в 512 біт на прості множники для алгоритму RSA та задачу логарифмування цілих чисел по 512 біт для алгоритму ElGamal, то друга задача є набагато складнішою від першої.

Однак є одна особливість. Якщо в системі, що побудована за допомогою алгоритму RSA, криптоаналітику вдалося розкласти відкритий ключ n одного з абонентів на два простих числа, то можливість зловживань обмежується тільки цим конкретним користувачем. У випадку ж системи, побудованої за допомогою алгоритму ElGamal, загрози розкриття зазнають всі абоненти криптографічної мережі.

Крім того, науковці Ленстра (Lenstra) і Манасс (Manasse) не тільки похитнули стійкість алгоритму RSA, розклавши в 1990 році дев'яте число Ферма на прості множники за досить короткий час [5], але й вказали на слабке місце у системі ElGamal, довівши, що підхід, який застосовується при розкладанні на прості множники дев'ятого числа Ферма, дозволяє для деяких спеціальних простих чисел суттєво вдосконалити методи дискретного логарифмування. Тобто той, хто вибирає для алгоритму ElGamal просте число p, має можливість вибрати спеціальне просте число, для якого задача дискретного логарифмування виявиться достатньо простою навіть для звичайних ЕОМ, не кажучи про сучасну потужну техніку. На сьогоднішній день відомі розклади на прості множники всіх чисел Ферма до F_{32} включно.

Однак ця проблема не є фатальною. Достатньо передбачити процедуру, яка буде гарантувати випадковість вибору простого p в системі ElGamal, і тоді факт заперечення криптостійкості алгоритму ElGamal втрачає силу. Слід зазначити, що чисел спеціального виду, які послабляють стійкість методу ElGamal, дуже мало, отже, випадковістю їхнього вибору можна знехтувати.

Експериментальні дослідження

Вибір параметрів досліджень. Для експериментальних досліджень вибрані наступні параметри алгоритмів шифрування:

1. Час генерування ключів.

За час генерування ключів приймається час, необхідний для визначення всіх відкритих і секретних ключів алгоритмом шифрування. Час генерування ключа алгоритмом залежить від розмірності (кількості біт) ключа. Він обчислюється в секундах або мілісекундах.

2. Час шифрування.

Часом шифрування вважається час, який алгоритму шифрування потрібно, щоб перетворити звичайний текст в зашифрований.

3. Час дешифрування.

Часом дешифрування вважається час, який алгоритму шифрування потрібно, щоб відтворити звичайний текст із зашифрованого тексту.

4. Пропускна здатність процесу шифрування.

Пропускна здатність процесу шифрування дорівнює кількості байт зашифрованого тексту, поділеній на час шифрування. Чим вища пропускна здатність, тим вищою буде продуктивність методу.

5. Пропускна здатність процесу дешифрування.

Пропускна здатність процесу дешифрування дорівнює кількості байт дешифрованого тексту, поділеній на час дешифрування.

6. Розмір зашифрованого файлу.

Розмір зашифрованого файлу дорівнює кількості байт зашифрованого тексту.

7. Розмір дешифрованого файлу.

Розмір дешифрованого файлу дорівнює кількості байт відтвореного тексту.

Експериментальні засоби та дані. Експерименти були проведені на Intel Core 2 Duo CPU процесор 2.09 ГГц з 4 ГБ оперативної пам'яті під операційною системою Windows 7.

Спочатку проводилось тестування для різних довжин ключів. У даній роботі кількість біт закритого ключа вибиралась у відповідності з NIST-рекомендаціями [6]. Відповідність розмірів ключів, що забезпечують еквівалентні рівні безпеки у алгоритмах RSA та ElGamal, показана у табл. 1. Ці п'ять конкретних рівнів безпеки були обрані тому, що вони являють собою п'ять відповідних рівнів роботи, необхідних для виконання пошуку ключа асиметричними алгоритмами шифрування: SKIPJACK, TRIPLE-DES, AES-малий, AES-середній та AES-великий відповідно [6]. Довжина повідомлення, яке використовувалось для шифрування, складала 105 КБ.

Таблиця 1

Відповідність розмірів закритих ключів у алгоритмах RSA та ElGamal

RSA, біт		ElGamal, біт	
10-ва система числення	2-ва система числення	10-ва система числення	2-ва система числення
1024	10000000000	160	10100000
2048	100000000000	224	11100000
3072	110000000000	256	100000000
7680	1111000000000	384	110000000
15360	11110000000000	512	1000000000

Далі проводилось тестування для різних розмірів вхідних файлів. При цьому розмір закритого ключа для алгоритму RSA був прийнятий рівним 1024 біт, а для алгоритму ElGamal відповідно 160 біт. Розміри текстових файлів, з якими проводились випробування були вибрані по 68 КБ, 105 КБ, 124 КБ і 235 КБ. Щоб досягти задовільного рівня достовірностей значень параметрів, кожна операція для кожного параметра тестування проводилась 20 разів і обчислювалось середнє його значення.

Результати тестувань. Результати проведених випробувань наведені у таблицях нижче. У табл. 2 і табл. 3 показані результати тестувань для різних довжин ключів алгоритмів RSA і ElGamal відповідно.

Таблиця 2

Результати тестування алгоритму RSA для різних довжин ключів

RSA ключ, біт	Час генерування ключів, с	Час шифрування, с	Час дешифрування, с
1024	1,312	0,202	10,082
2048	6,804	0,353	81,996
3072	32,108	0,378	196,925
7680	322,843	0,429	970,597
15360	н/в	н/в	н/в

Таблиця 3.

Результати тестування алгоритму ElGamal для різних довжин ключів

ElGamal ключ, біт	Час генерування ключів, с	Час шифрування, с	Час дешифрування, с
160	0,198	0,689	1,177
224	0,208	3,847	1,413
256	0,243	1,417	1,648
384	0,294	2,997	3,689
512	0,447	5,589	8,556

У табл. 4 і табл. 5 наведені результати тестувань для ключа одного розміру, але для різних розмірів вхідних файлів. Як правило, рекомендується для використання 1024-бітний ключ для алгоритму RSA і відповідний йому 160-бітний ключ для алгоритму ElGamal.

Таблиця 4

Результати тестування алгоритму RSA для різних розмірів вхідних файлів (1024-бітний ключ)

Розмір вхідного файлу, КБ	Час шифрування, с	Час дешифрування, с	Розмір зашифрованого файлу, КБ	Розмір дешифрованого файлу, КБ	Пропускна здатність шифрування, КБ/с	Пропускна здатність дешифрування, КБ/с
68	0,161	6,021	85,792	68	532,869	11,295
105	0,202	10,082	151,496	105	751,219	10,414
124	0,317	11,037	172,671	124	544,704	11,235
235	0,619	19,043	331,868	235	536,136	12,341

Таблиця 5

Результати тестування алгоритму ElGamal для різних розмірів вхідних файлів (160-бітний ключ)

Розмір вхідного файлу, КБ	Час шифрування, с	Час дешифрування, с	Розмір зашифрованого файлу, КБ	Розмір дешифрованого файлу, КБ	Пропускна здатність шифрування, КБ/с	Пропускна здатність дешифрування, КБ/с
68	0,475	0,404	136,003	68	286,481	168,477
105	0,689	1,177	210,010	105	304,827	89,181
124	0,743	1,319	249,001	124	335,058	94,0381
235	1,924	1,971	470,012	235	244,262	119,211

Аналіз результатів тестувань, проведених для різних розмірів ключів

Порівняння часу генерування ключів. В алгоритмах шифрування час генерування ключів є найважливішим суб-процесом, який вимагає генерування випадкових чисел і тестування їх на простоту. В алгоритмі RSA додаткового проводиться пошук цілого числа, взаємно простого зі значенням функції Ейлера. Це є достатньо трудомісткий процес. Він залежить від розміру ключа, але не залежить від розміру вхідних даних.

Для часу генерування ключів алгоритмом RSA були отримані значення у діапазоні від 1312 мс до 322843 мс, що відображено на рис. 1. Як правило, рекомендується для використання 1024-бітний ключ, час обчислення якого складає 1312 мілісекунд.

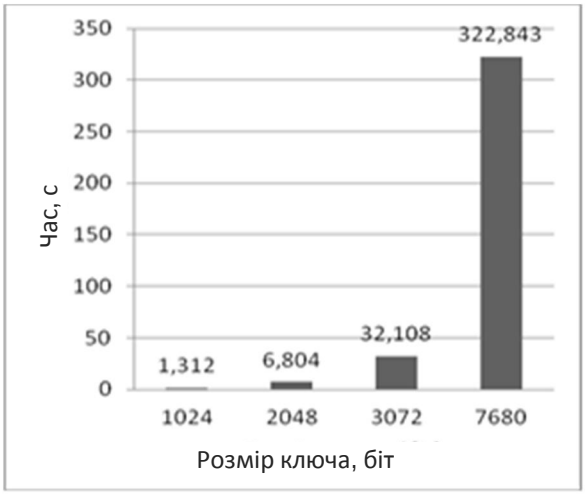

Рис. 1. **Залежність тривалості генерування ключа алгоритмом RSA від розміру ключа**

Час для генерації ключів алгоритмом ElGamal, крім генерування випадкових чисел і тестування їх на простоту, залежить тільки від розміру ключа.

Для часу генерування ключів алгоритмом ElGamal були отримані значення в діапазоні від 198 мс до 447 мс, що показано на рис. 2. Як правило, рекомендується для використання 160-бітний ключ, час обчислення якого складає 145 мілісекунд, що приблизно в 10 разів краще ніж в алгоритмі RSA.

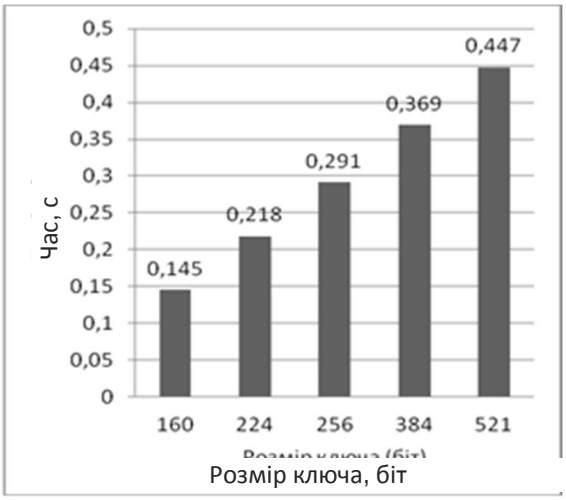

Рис. 2. **Залежність тривалості генерування ключа алгоритмом ElGamal від розміру ключа**

Порівняння часу шифрування. Як відомо, час, необхідний для шифрування з використанням швидкого піднесення до степеня, пропорційний числу одиничних біт у експоненті *e*. Тому зазвичай, у ролі ключа шифрування *e* беруть прості числа, що містять невелику кількість одиничних біт у двійковому записі, наприклад, прості числа Ферма 17, 257 або 65537. При дослідженні часу шифрування алгоритмом RSA в якості відкритого ключа e було прийнято значення e=65537, яке зазвичай рекомендується для ключа розмірністю 1024 біта для комерційного використання. При шифруванні 105-кілобайтного повідомлення для розмірів ключів у діапазоні від 1024 біт до 7680 біт були отримані результати у діапазоні від 0,202 с до 0,429 с, як показано на рис. 3.

При тестуванні алгоритму ElGamal при шифруванні 105-кілобайтного повідомлення для розмірів ключів у діапазоні від 160 біт до 384 біт були отримані результати у діапазоні від 0,689 с до 3,847 с (рис. 3).

Як видно з рис. 3 час шифрування повідомлення алгоритмом RSA кращий за час шифрування алгоритмом ElGamal для всіх розмірів ключів.

Порівняння часу дешифрування. У криптосистемах для полегшення операцій дешифрування застосовується китайська теорема про залишки, яка стверджує, що, якщо відомо розклад числа n на прості множники $n = n_1, n_2, \dots, n_k$, де всі n_i попарно взаємно прості числа, і результат приведення числа x за модулем $n_i \ \forall\, i = 1, \dots, k$ однаковий, то результатом приведення числа x за модулем n буде те ж число, тобто $\forall\, x$, a – цілих чисел, $x \equiv a \bmod n \Leftrightarrow x \equiv a \bmod n$ і $\forall\, i = 1, \dots, k$.

При дослідженні часу дешифрування алгоритмом RSA зашифрованого 105-кілобайтного повідомлення для розмірів ключів у діапазоні від 1024 біт до 7680 біт були отримані результати у діапазоні від 10,082 с до 970,597 с., як показано на рис. 4.

При дослідженні часу дешифрування алгоритмом ElGamal зашифрованого 105-кілобайтного повідомлення для розмірів ключів у діапазоні від 160 біт до 384 біт були отримані результати у діапазоні від 1,177 с до 3,689 с, що показано на рис. 4.

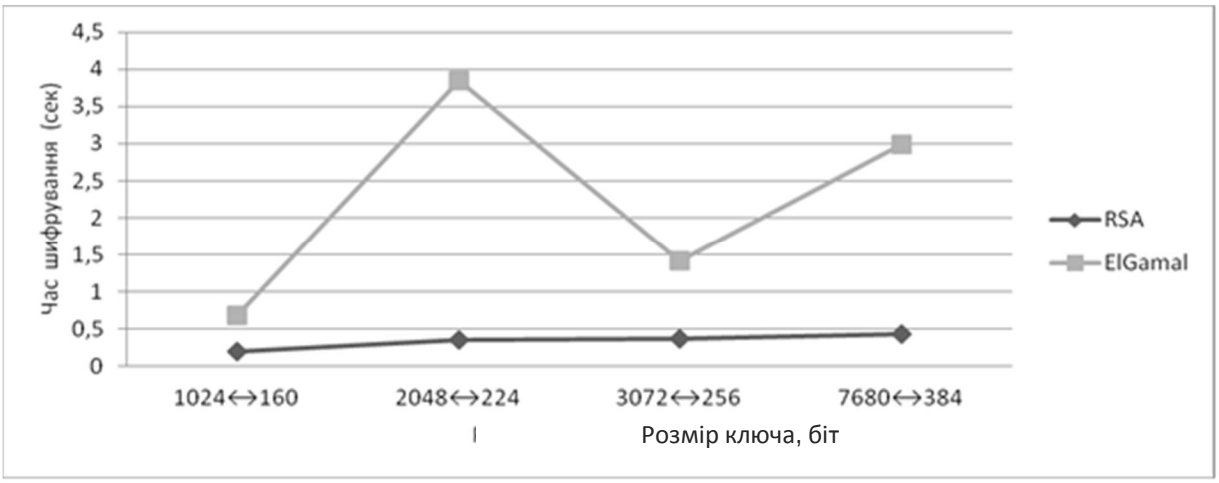

Рис. 3. **Графік залежності тривалості процесу шифрування від розміру ключа**

Аналіз показує, що обидва алгоритми мають майже однакові непогані результати при низькому рівні безпеки. Але при збільшенні розмірів ключів час дешифрування алгоритмом RSA зростає експоненціально, в той час, як час дешифрування алгоритмом ElGamal має лінійний порядок зростання

Аналіз результатів тестувань, проведених для різних розмірів вхідних повідомлень

Для подальших досліджень був прийнятий розмір ключа для алгоритму RSA рівний 1024 біт, а для алгоритму ElGamal відповідно 160 біт, що відповідає сучасним рекомендаціям з використання цих алгоритмів.

Розміри текстових файлів, з якими проводились, випробування були вибрані по 68, 105, 124 і 235 КБ.

Порівняння розмірів вихідних файлів. На рис. 5 і рис. 6 показані порівняння розмірів зашифрованих і дешифрованих файлів відповідно для алгоритмів RSA та ElGamal.

При шифруванні повідомлень алгоритмом RSA були отримані результати у діапазоні від 85,792 КБ до 331,868 КБ, а для алгоритму ElGamal у діапазоні від 136,003 КБ до 470,012 КБ. При дешифруванні повідомлень обома алгоритмами розміри дешифрованих файлів збігались з розмірами відповідних вхідних файлів.

Рис. 4. **Графік залежності тривалості процесу дешифрування від розміру ключа**

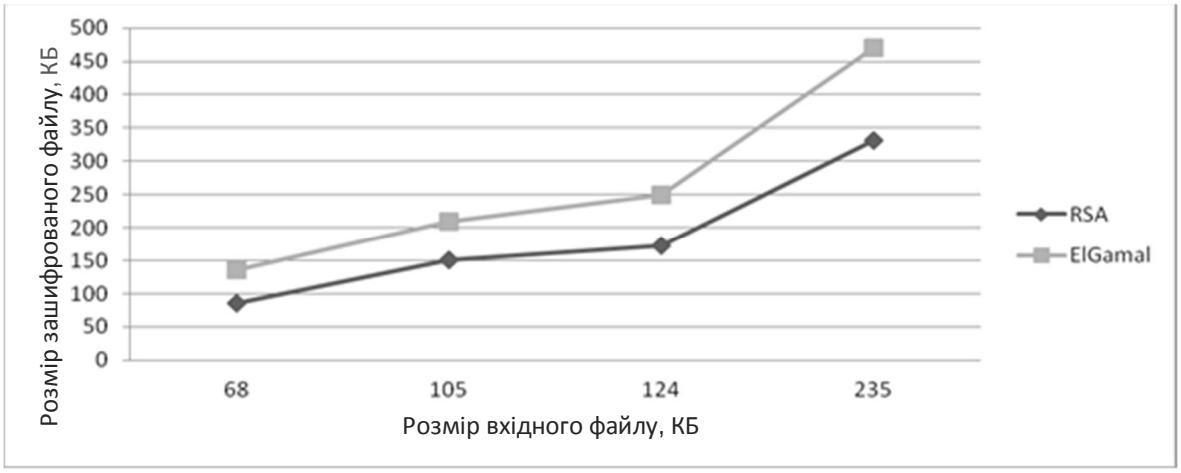

Рис. 5. **Графік залежності розміру зашифрованих даних від розміру вхідних даних**

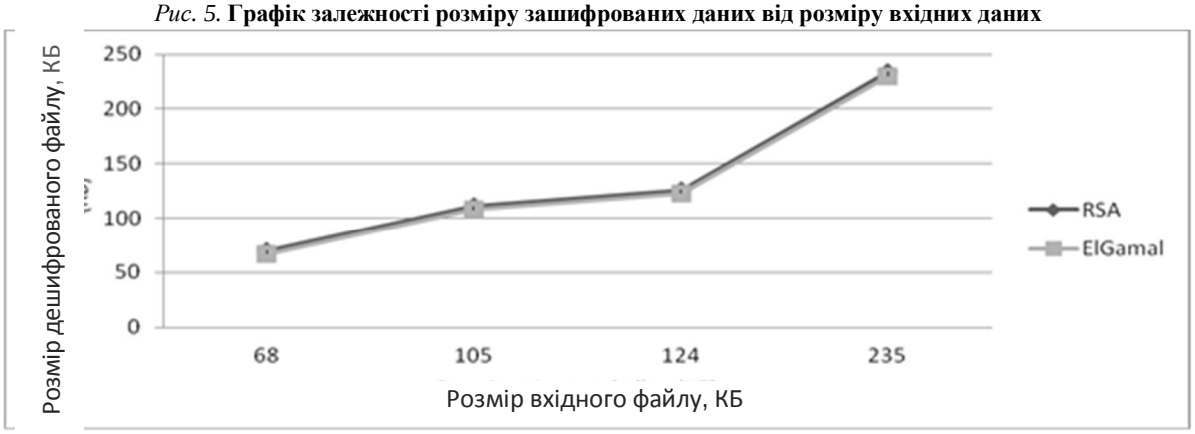

Рис. 6. **Графік залежності розміру дешифрованих даних від розміру вхідних даних**

При шифруванні алгоритмами RSA і ElGamal розмір зашифрованих даних залежить від розміру ключа та розміру вхідних даних.

Порівняння показали, що алгоритм RSA забезпечує кращу економію пропускного каналу. При цьому розмір зашифрованих даних більший за розмір вхідних даних в середньому на коефіцієнт 1,4. Довжина зашифрованих даних у алгоритмі ElGamal майже вдвічі довша вхідних даних.

Порівняння часу шифрування та часу дешифрування. При дослідженні часу шифрування алгоритмом RSA були отримані результати у діапазоні від 0,161 с до 0,619 с, як показано на рис. 7, а при дешифруванні цим алгоритмом – у діапазоні від 6,021 с до 19,043 с, як показано на рис. 8.

При тестуванні алгоритму ElGamal були отримані результати у діапазоні від 0,475 с до 1,924 с, як показано на рис. 7, а при дешифруванні цим алгоритмом – у діапазоні від 0,404 с до 1,971 с, як показано на рис. 8.

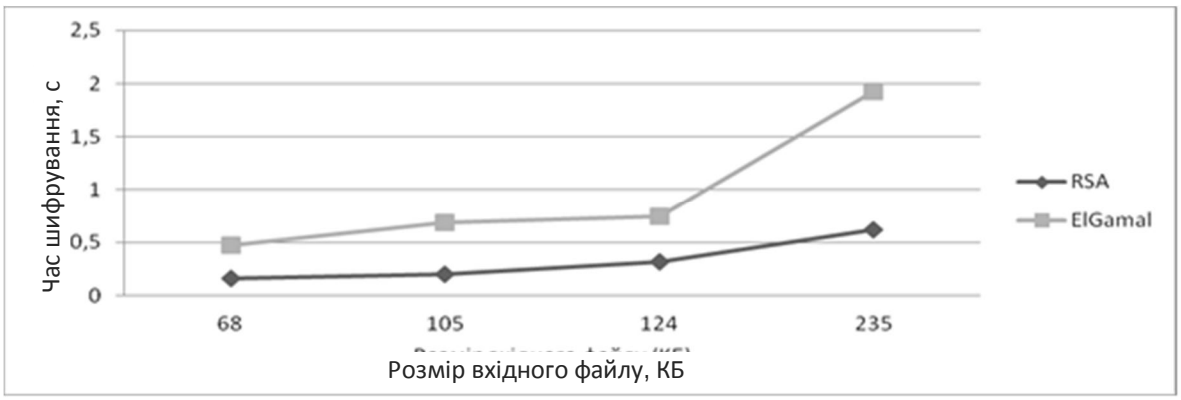

Рис. 7. **Графік залежності тривалості процесу шифрування від розміру вхідних даних**

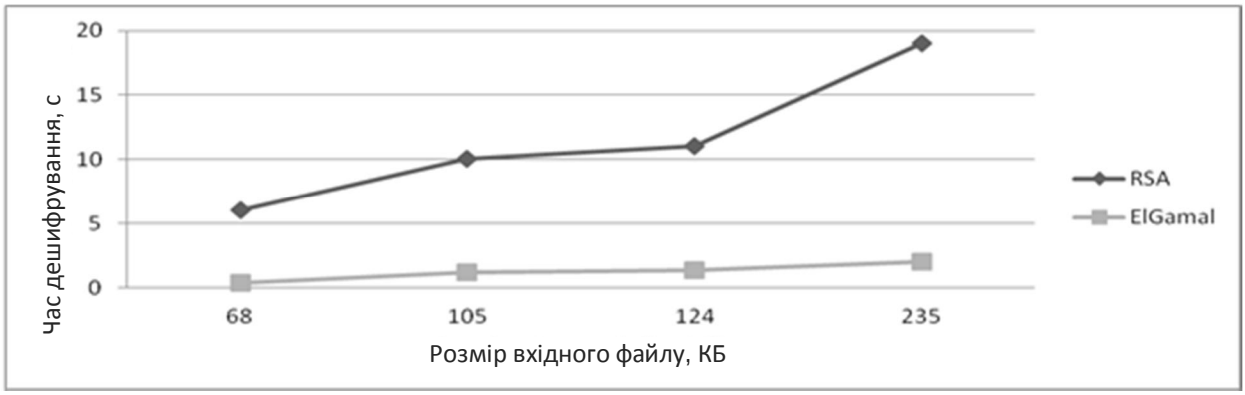

Рис. 8. **Графік залежності тривалості процесу дешифрування від розміру вхідних даних**

Порівняння тривалості процесів шифрування і дешифрування між алгоритмами показало, що алгоритм RSA має більш високу продуктивність під час шифрування, а алгоритм ElGamal кращий під час дешифрування.

Порівняння пропускної здатності алгоритмів. Пропускна здатність є найважливішим параметром, який демонструє ефективність будь-якого алгоритму. Рис. 9 демонструє пропускну здатність алгоритмів RSA і ElGamal для процесу шифрування та рис. 10 - для процесу дешифрування.

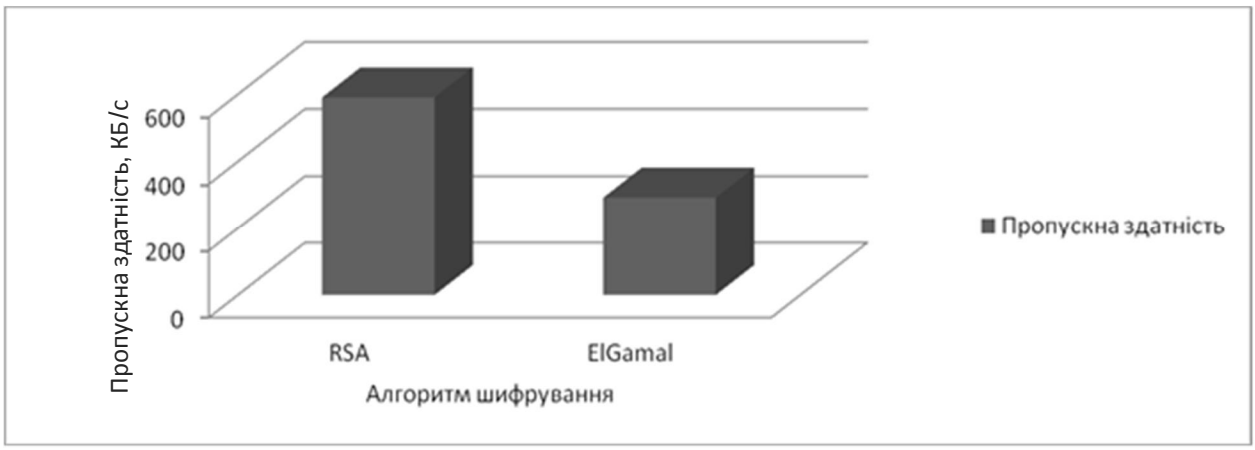

Рис. 9. **Пропускна здатність алгоритмів RSA і ElGamal в процесі шифрування**

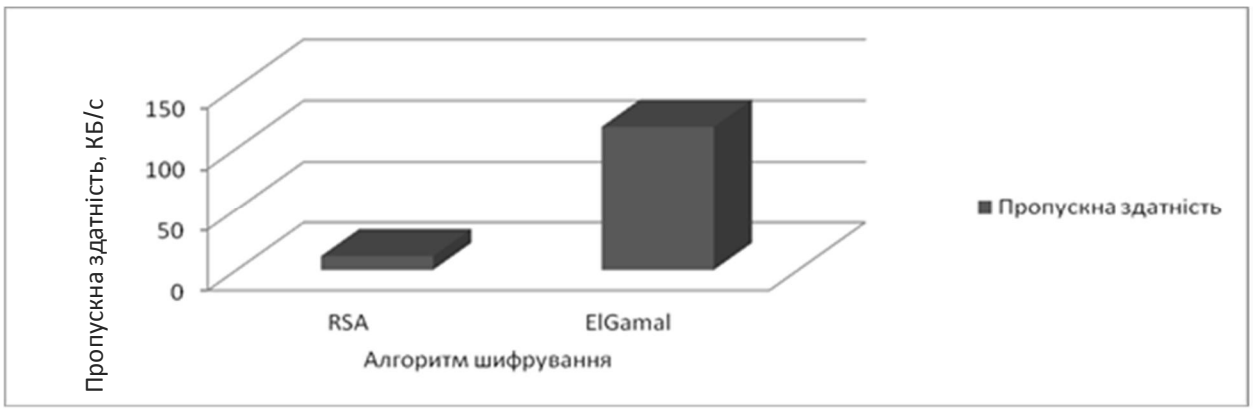

Рис. 10. **Пропускна здатність алгоритмів RSA і ElGamal в процесі дешифрування**

Як видно з цих діаграм, алгоритм RSA показав в 2 рази кращу пропускну здатність, ніж алгоритм ElGamal в процесі шифрування, проте алгоритм ElGamal показав в 10 разів кращу пропускну здатність у порівнянні з алгоритмом RSA в процесі дешифрування.

Висновки

Підсумовуючи проведені дослідження можна стверджувати:

1. Для всіх довжин ключів алгоритм ElGamal створює пару відкритого і закритого ключів в 10 разів швидше, ніж алгоритм RSA, що особливо помітно при значному збільшенні розмірів ключів.
2. Для алгоритму ElGamal час генерування ключів зростає лінійно зі збільшенням розміру ключа, в той час, як для алгоритму RSA він зростає в геометричній прогресії.
3. Час шифрування повідомлення алгоритмом RSA кращий, ніж алгоритмом ElGamal для всіх довжин ключів.
4. При дешифруванні обидва алгоритми показують майже однакові непогані результати при низькому рівні безпеки, але при збільшенні розмірів ключа час дешифрування алгоритмом RSA зростає експоненціально, тоді як час дешифрування алгоритмом ElGamal має лінійний порядок зростання.
5. Алгоритм RSA забезпечує кращу економію пропускного каналу. При цьому розмір зашифрованих алгоритмом RSA даних більший за розмір вхідних даних в середньому на коефіцієнт 1,4. Довжина зашифрованих даних у алгоритмі ElGamal вдвічі довша вхідних даних.
6. При дешифруванні повідомлень обома алгоритмами розміри дешифрованих файлів збігались з розмірами відповідних вхідних файлів.
7. Алгоритм RSA має більш високу швидкість під час шифрування, тоді як алгоритм ElGamal кращий під час дешифрування.
8. Алгоритм RSA показав в 2 рази кращу пропускну здатність, ніж алгоритм ElGamal в процесі шифрування, проте алгоритм ElGamal показав в 10 разів кращу пропускну здатність у порівнянні з алгоритмом RSA в процесі дешифрування.
9. Криптостійкість алгоритму ElGamal значно краща за криптостійкість алгоритму RSA.

Для підвищення швидкодії алгоритмів можна застосувати метод зменшення довжини ключа. Однак таке підвищення швидкодії може призвести до зменшення криптостійкості алгоритму. Такий підхід рекомендується використовувати, наприклад, при необхідності шифрування даних, які втрачають свою актуальність протягом короткого проміжку часу. У випадку неможливості зменшення криптостійкості пропонується підвищувати швидкодію за рахунок розпаралелювання обчислень в мультипроцесорних системах.

Доцільним є продовження роботи де даною темою із використанням вже готових результатів. Ця робота може стати основою або складовою більш масштабного проекту, для якого важливим фактором є збереження автентичності та захищеності інформації.

Список літератури

1. Пєвцов Г.В. Концептуальні підходи щодо забезпечення інформаційної безпеки у воєнній сфері / Г.В. Пєвцов, С.В. Залкін, А.О. Феклістов // Системи обробки інформації. – Х.: ХУПС, 2011. – Вип. 2 (92). – С. 57–59.

2. Вишняков В.М. Захист даних в інформаційних системах: навчальний посібник / В.М. Вишняков. – К.: КНУБА, 2010. – 128 с.

3. Ємець В.Ф. Сучасна криптографія / В.Ф. Ємець, А.О. Мельник, Р.Б. Попович. – Львів: БаК, 2003. – 149 с.

4. Шнайер Б. Прикладная криптография: Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Триумф, 2002. – 816 с.

5. Lenstra A.K., Manasse M.S. Factoring by Electronic Mail. Advances in Cryptology EUROCRYPT '89 Proceedings. Springcr-Verlag,1990. pp. 355-371.

6. National Institute of Standards and Technology Special Publication 800-124 Revision 1 Natl. Inst. Stand. Technol. Spec. Publ. 800-124 Rev. 1, 29 pages (June 2013) URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-124r1.pdf (Last accessed: 07.02.2019).

Рецензент: доктор технічних наук, професор Ю.В. Шабатура, Національна академія сухопутних військ імені гетьмана Петра Сагайдачного, Львів.

ИССЛЕДОВАНИЯ И СРАВНИТЕЛЬНЫЙ АНАЛИЗ АЛГОРИТМОВ АСИММЕТРИЧНОГО ШИФРОВАНИЯ

Т.А. Матвейчук, В.Д. Смычок

В работе с помощью разработанного программного продукта были исследованы и проведен сравнительный анализ асимметричных алгоритмов шифрования текстовых данных, их преимуществ и недостатков, криптографической стойкости, дана экспериментальная оценка их характеристик относительно эффективности использования ими памяти компьютера, продолжительности процессов генерации ключей, зашифровывания и расшифровывания данных, пропускной способности алгоритмов, размерностей ключей, объемов зашифрованных и расшифрованных файлов. На основании полученных результатов даны рекомендации по применению рассмотренных методов шифрования. В качестве асимметричных алгоритмов шифрования выбраны алгоритмы RSA и ElGamal.

С точки зрения практической реализации, как программным, так и аппаратным способом, ощутимой разницы между этими двумя методами нет. Однако в своих характеристиках они заметно отличаются. Коротко, полученные основные оценочные результаты заключаются в следующем: криптостойкость алгоритма ElGamal значительно выше криптостойкости алгоритма RSA; алгоритм RSA имеет более высокую скорость при зашифровывании информации, а алгоритм ElGamal лучший во время расшифровывания; при увеличении размеров ключей время расшифровки алгоритмом RSA растет экспоненциально, в то время, как длительность расшифровки алгоритмом ElGamal имеет линейный порядок роста; алгоритм RSA показал в 2 раза лучшую пропускную способность чем алгоритм ElGamal в

процессе зашифровывания информации, зато алгоритм ElGamal показал в 10 раз лучшую пропускную способность по сравнению с алгоритмом RSA в процессе расшифровывания информации; длина зашифрованных данных алгоритмом ElGamal в 2 раза длиннее исходных данных, в то время как размер зашифрованных алгоритмом RSA данных больше размера исходных данных в среднем на коэффициент 1,4; для всех длин ключей алгоритм ElGamal создает пару открытого и закрытого ключей в среднем в 10 раз быстрее, чем алгоритм RSA, что особенно заметно при значительном увеличении размеров ключей; для алгоритма ElGamal время генерирования ключей растет линейно с увеличением размеров ключей, в то время как для алгоритма RSA оно растет в геометрической прогрессии.

Ключевые слова: *асимметричные алгоритмы, шифрование, дешифрование, криптостойкость, RSA, ElGamal.*

RESEARCH AND COMPARATIVE ANALYSIS OF ALGORITHMS OF ASYMMETRIC ENCRYPTION

T. Matveichuk, V. Smychok

Using a developed software product, a comparative analysis of asymmetric text data encryption algorithms, their advantages and disadvantages, cryptographic strength, experimental evaluation of their characteristics with respect to the efficiency of their computer memory usage, the duration of key generation, data encryption and decryption ability of algorithms, key dimensions, volumes of encrypted and decrypted files has been made. Based on the results obtained, recommendations were given on the use of the encryption methods considered. The RSA and ElGamal algorithms are chosen as asymmetric encryption algorithms.

From the point of view of practical implementation, both in software and hardware, there is no tangible difference between these two methods. However, in their characteristics they are noticeably different. Briefly, the main estimated results obtained are as follows: the cryptographic strength of the ElGamal algorithm is significantly higher than that of the RSA algorithm; the RSA algorithm has a higher speed when encrypting information, and the ElGamal algorithm has the best during decryption; as the key sizes increase, the decryption time by the RSA algorithm grows exponentially, while the duration of the decryption by the ElGamal algorithm has a linear growth order; the RSA algorithm showed 2 times better bandwidth than the ElGamal algorithm in the process of encoding information, but the ElGamal algorithm proved 10 times better throughput compared to the RSA algorithm in the process of decrypting information; the length of the encrypted data by the ElGamal algorithm is 2 times longer than the original data, while the size of the data encrypted by the RSA algorithm is larger than the size of the original data by an average of 1.4; for all key lengths, the ElGamal algorithm creates a pair of public and private keys on average 10 times faster than the RSA algorithm, which is especially noticeable with a significant increase in key sizes; for the ElGamal algorithm, the key generation time increases linearly with increasing key sizes, while for the RSA algorithm it grows exponentially.

Keywords*: asymmetric algorithms, encryption, decryption, cryptographic resistance, RSA, ElGamal*.

УДК 358.1 DOI: https://doi.org/10.33577/2312-4458.20.2019.28-32

Д.А. Новак, І.Д. Волков

Науково-дослідний центр ракетних військ і артилерії, м. Суми

МЕТОДИЧНИЙ ПІДХІД ДО ВИЗНАЧЕННЯ ЗАЛЕЖНОСТІ ВІДХИЛЕННЯ ПОЧАТКОВОЇ ШВИДКОСТІ СНАРЯДІВ ЧЕРЕЗ ПОДОВЖЕННЯ ЗАРЯДНОЇ КАМОРИ АРТИЛЕРІЙСЬКИХ ГАРМАТ, ВИМІРЯНЕ ЗА ДОПОМОГОЮ ПЗК

Моніторинг технічного стану артилерійського озброєння Збройних Сил України свідчить про достатній ступінь зносу каналів стволів для значної кількості артилерійських систем, що безпосередньо впливає на ефективність виконання вогневих завдань артилерією. У статті наведено порядок та особливості визначення відхилення початкових швидкостей снарядів через подовження зарядної камори артилерійських гармат з використанням таблиць внутрішньої балістики; надано інформацію про взаємозв'язок між основними параметрами внутрішньої балістики та їх впливом на значення початкової швидкості снарядів; визначено перелік вхідних даних, необхідних для визначення рівня падіння початкової швидкості снарядів для артилерійських систем з відомим рівнем зносу каналу ствола.

Ключові слова: *внутрішня балістика, артилерійська система, початкова швидкість снарядів, знос каналу ствола, подовження зарядної камори артилерійської гармати.*