УДК 004.056.55
DOI https://doi.org/10.32782/tnv-tech.2023.3.2

РЕАЛІЗАЦІЯ КРИПТОСТІЙКОГО АЛГОРИТМУ ІЗ ПРОСТОЮ ПРОЦЕДУРОЮ ШИФРУВАННЯ ТА ДЕШИФРУВАННЯ НА ОСНОВІ ЕЛІПТИЧНИХ КРИВИХ

Завгородній В. В. *– доктор технічних наук, професор,*
завідувач кафедри інформаційних технологій
Державного університету інфраструктури та технологій
ORCID ID: 0000-0002-8347-7183

Завгородня Г. А. *– кандидат технічних наук, доцент,*
доцент кафедри інформаційних технологій
Державного університету інфраструктури та технологій
ORCID ID: 0000-0001-8523-1761

Березінський Ю. С. *– магістр кафедри інформаційних технологій*
Державного університету інфраструктури та технологій
ORCID ID: 0009-0001-7745-0080

Березінська І. П. *– магістр кафедри інформаційних технологій*
Державного університету інфраструктури та технологій
ORCID ID: 0009-0006-3224-1505

У роботі розглядаються різні методи та переваги застосування криптографічних алгоритмів, що базуються на теорії еліптичних кривих. Також наведені приклади реалізації схем електронно-цифрового підпису, які базуються на цих алгоритмах. Однією з ключових переваг алгоритмів на основі еліптичних кривих є забезпечення високого рівня безпеки при менших розмірах ключів порівняно з іншими криптографічними системами.

У роботі детально розглядається, які саме еліптичні криві можуть бути використані в різних криптографічних схемах. Також наводиться порівняння з іншими алгоритмами, такими як RSA чи DSA, зазначаючи переваги використання еліптичних кривих.

У роботі наведено приклади реалізації схем електронно-цифрового підпису з використанням еліптичних кривих. Електронно-цифровий підпис є важливим механізмом для забезпечення автентифікації та цілісності даних у криптографії, і використання еліптичних кривих може підвищити ефективність та безпеку таких підписів.

Під час тестування було виявлено лінійну залежність часу від кількості запусків алгоритмів. Час виконання алгоритму RSA приблизно дорівнює часу виконання алгоритму DSA. Це пов'язано, в першу чергу, з лінійним запуском алгоритмів. Коли мова йде про схеми, що ґрунтуються на еліптичних кривих, вони дозволяють досягти бажаного рівня безпеки за значно меншої довжини ключа, ніж у випадку з схемою RSA. При використанні еліптичних кривих можна забезпечити однаковий рівень захисту інформації, використовуючи ключі меншого розміру порівняно з традиційними методами, такими як RSA.

Робота допомагає розібратися з використанням теорії еліптичних кривих у криптографії, надаючи інформацію про різні методи та застосування цих алгоритмів, зокрема у схемах ЕЦП.

Ключові слова: *еліптичні криві, електронно-цифровий підпис, шифрування, дешифрування, алгоритм, кубика.*

Zavgorodnii V. V., Zavgorodnya A. A., Berezinsky Yu. S., Berezinska I. P. Implementation of a cryptographically strong algorithm with a simple encryption and decryption procedure based on elliptic curves

The work considers various methods and advantages of using cryptographic algorithms based on the theory of elliptic curves. Examples of implementation of electronic digital signature

schemes based on these algorithms are also given. One of the key advantages of algorithms based on elliptic curves is to provide a high level of security with smaller key sizes compared to other cryptographic systems.

The work examines in detail which elliptic curves can be used in various cryptographic schemes. A comparison with other algorithms such as RSA or DSA is also given, noting the advantages of using elliptic curves.

The paper gives examples of implementation of electronic digital signature schemes using elliptic curves. Digital signatures are an important mechanism for authentication and data integrity in cryptography, and the use of elliptic curves can improve the effectiveness and security of such signatures.

During testing, a linear dependence of time on the number of algorithm runs was revealed. The execution time of the RSA algorithm is approximately equal to the execution time of the DSA algorithm. This is primarily due to the linear launch of the algorithms. When it comes to schemes based on elliptic curves, they allow you to achieve the desired level of security with a much smaller key length than in the case of the RSA scheme. When using elliptic curves, it is possible to provide the same level of information protection by using smaller keys compared to traditional methods such as RSA.

The work helps to understand the use of the theory of elliptic curves in cryptography, providing information on various methods and applications of these algorithms, in particular, in electronic digital signature schemes.

Key words: *elliptic curves, digital signature, encryption, decryption, algorithm, cubic plane curve.*

Постановка проблеми. У сучасному світі значення інформації постійно зростає, інформатизація суспільства постійно зростає. Це призводить до необхідності вдосконалювати методи та засоби захисту інформації.

До захищених інформаційних систем пред'являються ряд особливих вимог, які випливають з властивостей інформації: конфіденційності, доступності та цілісності. Найбільш популярним методом захисту інформації є використання криптографічних алгоритмів [1].

З метою забезпечення захисту інформації використовуються наступні криптографічні примітиви: симетричні криптосистеми, асиметричні криптосистеми, цифрові підписи, криптографічні хеш-функції та коди перевірки автентичності [2].

Ці методи допомагають забезпечити захист інформації та полегшують безпечний обмін даними в сучасному інформаційному світі.

Аналіз останніх досліджень і публікацій. З розвитком фінансової та комерційної сфери діяльності все важливішою стає роль засобів та систем криптографічного захисту інформації [3]. Це зумовлено не лише необхідністю переходу до «електронної основи», а й значним розширенням можливостей передачі, обробки та зберігання інформації в розподілених обчислювальних системах. Використання спеціальних криптографічних протоколів та криптосистем дозволяє здійснювати різноманітні економічні взаємовідносини на відстані, що виключає необхідність особистих зустрічей між учасниками, забезпечуючи при цьому необхідну фінансову та правову дисципліну. Прикладом такої взаємодії є використання електронно-цифрового підпису [4].

Для підвищення криптостійкості алгоритму цифрового підпису рекомендується використовувати алгоритми, засновані на еліптичних кривих.

Еліптичні криві – це криві першого роду з раціональними точками [5]. Раціональні криві (в алгебраїчно замкнутому полі) представляють собою алгебраїчні криві роду 0. Алгебраїчна крива – це геометричне місце (множина) точок на площині, визначене як набір розв'язків многочлена від двох змінних. Зазвичай алгебраїчні криві мають розмірність 1. Такі криві – це алгебраїчні різноманіття, всі підмножини яких складаються з однієї точки [4].

Кожну таку криву можна представити у вигляді кубики без особливостей [6]. Кубика – це плоска крива третього порядку. Всі точки площини, які задовольняють кубічному рівнянню в однорідних координатах на проективній площині, складають множину точок кубики $F(x, y, z) = 0$.

Еліптична крива виникає в результаті перетину двох конічних перетинів – перетину площини з круговим конусом [7]. Такий перетин є кривою четвертого порядку роду 1 і, отже, є еліптичною кривою, якщо містить хоча б одну раціональну точку. У випадку відсутності раціональної точки, перетин може бути раціональною кривою четвертого порядку з особливостями, або розкладатися на криві меншого порядку.

У сучасній криптографії актуальним є питання підвищення стійкості та зменшення розмірів блоків даних шляхом модифікації вже існуючих криптосистем. Найочевиднішим вирішенням зазначеної проблеми є представлення блоків інформації в криптографічних алгоритмах не лише у вигляді чисел або елементів скінченних полів, але й у вигляді інших алгебраїчних об'єктів більшої складності. Один з відповідних типів таких об'єктів – точки на еліптичних кривих [7].

Спочатку термін «еліптична крива» позначав гладку криву на декартовій площині, що описувалася наступним рівнянням:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

Проте в контексті криптографії, еліптичні криві використовуються для вирішення різних задач, таких як генерація ключів, підписи повідомлень та шифрування даних. Такий підхід дозволяє досягти вищого рівня безпеки та ефективності порівняно з традиційними методами криптографії, що базуються на інших математичних структурах [5].

Якщо всі коефіцієнти і невідомі – дійсні числа, то шляхом заміни змінних рівняння може бути перетворено до більш простого вигляду:

$$y^2 = x^3 + ax + b \quad (2)$$

Еліптичні криві поділяються на два типи: сингулярні та несингулярні.

Для несингулярних еліптичних кривих виконується наступна нерівність:

$$y^2 = x^3 + ax^2 + b \quad (3)$$

Однак для сингулярних кривих ця умова не виконується.

У схемах електронного цифрового підпису не рекомендується використовувати сингулярні криві [4]. Використання таких кривих може значно знизити стійкість схеми електронного цифрового підпису.

Замість цього, бажано використовувати еліптичні криві без особливих точок. Геометрично це означає, що на графіку кривої не повинно бути точок самоперетину чи повернення. Алгебраїчно, це досягається тим, що дискримінант кривої відрізняється від нуля:

$$\Delta = -16\left(4a^3 + 27b^2\right) \quad (4)$$

Якщо у еліптичної кривої немає особливих точок і дискримінант дорівнює додатному значенню, то на графіку присутні дві зв'язні компоненти. Якщо у кривої немає особливих точок і дискримінант від'ємний, то на графіку буде присутня лише одна компонента.

Формулювання цілей статті. Мета статті – вивчення теорії еліптичних кривих та реалізація достатньо криптостійкого алгоритму з простою процедурою шифрування та дешифрування на основі еліптичних кривих.

Виклад основного матеріалу. Розглянемо простий підхід до шифрування та дешифрування, використовуючи еліптичні криві. Метою завдання є зашифрування повідомлення Mes, яке може бути представлене як точка на еліптичній кривій $Dot_{Mes}(x, y)$. Аналогічно до обміну ключами, система шифрування та дешифрування використовує параметри еліптичної кривої $ElCur(a, b)$ та точку $Point$ на ній. Учасник $Mem2$ обирає закритий ключ $ClKey$ та обчислює відкритий ключ $OpKey = ClKey \times Point$. Для дешифрування повідомлення $Dot_{Mes}(x, y)$ використовується відкритий ключ одержувача $Mem2$. Учасник $Mem1$ також обирає випадкове ціле додатне число $posNum$ та обчислює зашифроване повідомлення Enc_{Mes}, яке представляє точку $DotPoint$ на еліптичній кривій $ElPoint$:

$$Enc_{Mes} = \{posNum \times Point, Dot_{Mes} + DotPoint \times ElPoint\} \tag{5}$$

Для дешифрування повідомлення, одержувач $Mem2$ множить першу координату точки на свій закритий ключ і віднімає результат від другої координати:

$$\begin{aligned} &Dot_{Mes} + posNum \times OpKey - ClKey \times (poxNum \times Point) = \\ &= Dot_{Mes} + poxNum \times (ClKey \times Point) - ClKey \times (posNum \times Point) = Dot_{Mes} \end{aligned} \tag{6}$$

Наприклад, учасник $Mem1$ зашифровує повідомлення Dot_{Mes}, додаючи до нього $posNum \times OpKey$. При цьому значення ключа $posNum$ залишається невідомим, тому ніхто не може його знайти, навіть якщо $OpKey$ є відкритим ключем. Учаснику $Mem2$ для відновлення повідомлення необхідно обчислити $posNum$. Це завдання дуже складне для обчислення.

Одержувач також не знає ключ $posNum$, але може скористатися підказкою $posNum \times Point$. Помноживши $posNum \times Point$ на свій закритий ключ, він отримає значення, яке було додане відправником. Таким чином, одержувач, не знаючи ключа $posNum$, але маючи свій закритий ключ, може відновити незашифроване повідомлення.

Для вирішення завдань автентифікації та забезпечення цілісності інформації використовується концепція електронного цифрового підпису (ЕЦП). ЕЦП – це набір методів, що дозволяють перенести властивості рукописного підпису до електронного документообігу. Головною відмінністю цифрового підпису є можливість неодноразового його копіювання, що створює необхідність вирішувати дану задачу математичними методами [6].

Існує безліч варіантів реалізації електронно-цифрового підпису, але найбільш відомими є алгоритми *RSA* (*Rivest*, *Shamir*, *Adleman*) та *DSA* (*Digital Signature Algorithm*). Розглянемо схему формування ЕЦП на їх основі.

Схема формування ЕЦП на основі алгоритму *RSA* включає такі кроки:

1. Генерація ключів:

Крок 1: Вибір двох великих простих чисел p та q.

Крок 2: Обчислення добутку $n = p \times q$. Це буде модуль для відкритого та закритого ключів.

Крок 3: Обчислення функції Ейлера для n: $\varphi(n) = (p-1) \times (q-1)$.

Крок 4: Вибір відкритої експоненти e, такої, що $1 < e < \varphi(n)$ та e взаємно просте з $\varphi(n)$. Зазвичай обирають значення $e = 65537$.

Крок 5: Обчислення закритої експоненти d, оберненої до e за модулем $\varphi(n)$, тобто $d \equiv e^{-1} \, mod \, \varphi(n)$. Це можна зробити за допомогою розширеного алгоритму Евкліда.

2. Формування ЕЦП:

Нехай у нас є повідомлення *Mes*, яке потрібно підписати за допомогою закритого ключа (d,n). ЕЦП формується наступним чином:

Крок 1: Обчислення хеш-значення повідомлення *Mes* з використанням хеш-функції (наприклад, SHA-256). Нехай $H(Mes)$ – це хеш-значення.

Крок 2: Шифрування хеш-значення за допомогою закритого ключа: $S = (H(Mes))^d \, mod \, n$. Отримане значення S і буде ЕЦП для повідомлення *Mes*.

3. Перевірка ЕЦП:

Для перевірки ЕЦП отримувачем з використанням відкритого ключа (e,n) виконуються наступні кроки:

Крок 1: Отримання ЕЦП S та початкового повідомлення *Mes*.

Крок 2: Обчислення хеш-значення повідомлення *Mes*: $H(Mes)$.

Крок 3: Розшифрування ЕЦП за допомогою відкритого ключа: $S^e \, mod \, n$.

Крок 4: Порівняння отриманого розшифрованого значення з хеш-значенням $H(Mes)$. Якщо вони співпадають, то ЕЦП вважається вірним.

Алгоритм *RSA* має обмеження на розмір ключів, і для забезпечення безпеки ключі повинні бути достатньо великими (розміром 2048 біт і більше). Також важливо звертатися до перевірених бібліотек та програмних реалізацій *RSA*, щоб уникнути можливих вразливостей.

Схема формування ЕЦП на основі алгоритму *DSA* включає такі кроки:

1. Генерація ключів:

Крок 1: Генерація параметрів *DSA*:

– Обирається просте число q, яке буде використовуватись як порядок підгрупи групи точок еліптичної кривої (для більш безпечних варіантів алгоритму) або як модуль для обчислення $p = k \times q + 1$.

– Обирається просте число p, яке визначає порядок групи точок еліптичної кривої.

– Обирається ціле число g, яке є генератором групи точок еліптичної кривої.

Крок 2: Генерація ключів:

– Обирається випадкове ціле число x (закритий ключ) з інтервалу $[1, q-1]$.

– Обчислюється відкритий ключ y як $y = g^x \, mod \, p$.

2. Підписування повідомлення:

Припустимо, що у нас є повідомлення *Mes*, яке потрібно підписати.

Крок 1: Обирається випадкове ціле число k з інтервалу $[1, q-1]$.

Крок 2: Обчислюється точка еліптичної кривої r як $r = (g^k \, mod \, p) \, mod \, q$.

Крок 3: Обчислюється s як $s = (k^{-1} \times (H(Mes) + x \times r)) \, mod \, q$, де $H(Mes)$ – хеш повідомлення *Mes* (зазвичай використовується хеш-функція, наприклад, *SHA*-256).

Крок 4: Пара значень (r,s) представляє ЕЦП для повідомлення *Mes*.

3. Перевірка ЕЦП:

Для перевірки підпису потрібний відкритий ключ отримувача y, повідомлення *Mes* та пара значень (r,s) ЕЦП.

Крок 1: Обчислюється хеш повідомлення $H(Mes)$.

Крок 2: Обчислюється w як $w = s^{-1} \, mod \, q$.

Крок 3: Обчислюється $u1$ як $u1 = \left(H\left(Mes\right) \times w\right) mod\, q$, а $u2 = \left(r \times w\right) mod\, q$.

Крок 4: Обчислюється точка еліптичної кривої v як $v = \left(\left(g^{u1} \times y^{u2}\right) mod\, p\right) mod\, q$.

Крок 5: Якщо v дорівнює r, підпис вважається правильним, інакше він вважається недійсним.

Безпека алгоритму *DSA* тісно пов'язана з правильним вибором параметрів q, p та g, а також з правильною генерацією випадкових чисел k та x. Будь-які порушення або помилки на цих етапах можуть призвести до компрометації підпису та загрози безпеці даних.

Для реалізації алгоритмів була обрана мова програмування *Python*. Алгоритми бібліотеки *pygost* мови програмування *Python* базуються на еліптичних кривих.

Для тестування програмного продукту був обраний інструмент *Unittest*. Це стандартний модуль для написання юніт-тестів на *Python*. *Unittest* є портом *JUnit* з *Java*. Іншими словами, як у коді модуля, так і при написанні тестів легко прослідковується об'єктно-орієнтований стиль програмування, що дуже зручно для тестування процедур і класів [8].

Для порівняння часу виконання алгоритмів *RSA* і *DSA* був побудований графік часу виконання алгоритмів (рис. 1). В якості вихідних даних використовувались результати тестування виконання алгоритмів. Алгоритми запускалися в одному потоці по одному процесу. Кількість запусків алгоритму збільшувалась з кожним кроком на 100.

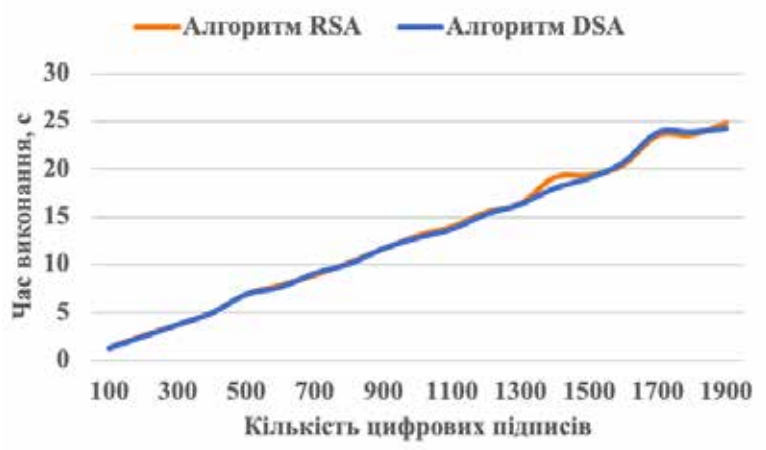

Рис. 1. Графік часу виконання алгоритмів

За результатами тестування була сформована таблиця порівняння алгоритмів електронно-цифрового підпису (табл. 1).

За результатами тестування очевидна лінійна залежність часу виконання від кількості запусків алгоритму. Зі збільшенням кількості запусків зростає час виконання. Також можна помітити, що час виконання алгоритму *RSA* приблизно дорівнює часу виконання алгоритму *DSA*. Це пов'язано, перш за все, з швидкодією байт-коду *Python*.

Таблиця 1

Порівняння алгоритмів електронно-цифрового підпису

Кількість цифрових підписів	**Алгоритм *RSA*, с**	**Алгоритм *DSA*, с**
100	1,25	1,27
200	2,56	2,46
300	3,7	3,72
400	4,96	4,94
500	6,88	6,88
600	7,82	7,63
700	8,91	9,09
800	10,2	10,08
900	11,64	11,66
1000	13,02	12,8
1100	13,97	13,71
1200	15,5	15,27
1300	16,38	16,31
1400	19,16	17,98
1500	19,38	19,04
1600	20,46	20,69
1700	23,53	23,82
1800	23,61	23,87
1900	24,84	24,25

Висновки. У даній роботі викладено базові поняття теорії еліптичних кривих, необхідні для реалізації криптографічних протоколів. Розглянуті алгоритми шифрування *RSA*, *DSA* та алгоритми створення електронно-цифрового підпису з використанням еліптичних кривих. Результатом цієї роботи стали приклади реалізації схеми ЕЦП *RSA* та *DSA* на мові *Python*. Ці алгоритми реалізовані достатньо криптостійко з простою процедурою шифрування та дешифрування.

На основі проведеної роботи можна виділити основні переваги еліптичної криптографії: у криптографії, що базується на еліптичних кривих, довжина ключа значно менша порівняно з іншими алгоритмами асиметричної криптографії. Еліптичні алгоритми працюють набагато швидше, ніж класичні. Це можна пояснити розміром ключа та використанням структури бінарного скінченного поля. Завдяки малій довжині ключа та високій швидкості роботи, алгоритми на еліптичних кривих можуть застосовуватись у сім-картах та інших пристроях із обмеженими обчислювальними ресурсами. В результаті аналізу алгоритмів вирішення задачі дискретного логарифмування було виявлено, що зламати алгоритм шифрування на основі еліптичних кривих досить складно, якщо підібрані правильні параметри. До недоліків можна віднести проблему вибору відповідної еліптичної кривої та проблему, пов'язану з генерацією ключів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Timothy P. Layton. Information Security. *Auerbach Publications*. 2016. P. 264. ISBN 9781420013412

2. John R. Vacca. Computer and Information Security handbook. *Morgan Kaufmann*. 2017. eBook ISBN 9780128039298

3. Ali Evren Göksungur. Electronic Signature and Electronic Document Management Systems. *Scholars' Press*. 2018. P. 56. ISBN 9783330652200

4. Elaine Barker, Lily Chen, Allen Roginsky, Miles Smid. Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography. *National Institute of Standards and Technology*, 2013. ISBN 1495447502.

5. Avani Shah, Vinayak Bharadi. Online Signature Recognition Using Sectorization of Complex Plane. *LAP LAMBERT Academic Publishing*. 2014. P. 104. ISBN 9783659594199

6. Nivethaa Shree, Latha Parthiban. Elliptic Curve Cryptography for Digital Signature Authentication. *LAP LAMBERT Academic Publishing*. 2013. P. 60. ISBN 9783659263958

7. Martin Krisell. Elliptic Curve Digital Signatures in RSA Hardware. *Scholars' Press*. 2013. P. 108. ISBN 9783639511826

8. Fabrizio Romano, Heinrich Kruger. Learn Python Programming. *Packt Publishing*. 2023. P. 552. ISBN 9781801815529

REFERENCES:

1. Timothy P. Layton. (2016) Information Security. *Auerbach Publications*. P. 264. ISBN 9781420013412

2. John R. Vacca. (2017) Computer and Information Security handbook. *Morgan Kaufmann*. eBook ISBN 9780128039298

3. Ali Evren Göksungur. (2018) Electronic Signature and Electronic Document Management Systems. *Scholars' Press*. P.56. ISBN 9783330652200

4. Elaine Barker, Lily Chen, Allen Roginsky, Miles Smid. (2013) Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography. *National Institute of Standards and Technology*. ISBN 1495447502.

5. Avani Shah, Vinayak Bharadi. (2014) Online Signature Recognition Using Sectorization of Complex Plane. *LAP LAMBERT Academic Publishing*. P. 104. ISBN 9783659594199

6. Nivethaa Shree, Latha Parthiban. (2013) Elliptic Curve Cryptography for Digital Signature Authentication. *LAP LAMBERT Academic Publishing*. P. 60. ISBN 9783659263958

7. Martin Krisell. (2013) Elliptic Curve Digital Signatures in RSA Hardware. *Scholars' Press*. P. 108. ISBN 9783639511826

8. Fabrizio Romano, Heinrich Kruger. (2023) Learn Python Programming. *Packt Publishing*. P. 552. ISBN 9781801815529