НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації

«На правах рукопису»
УДК 003.26:519.688

«До захисту допущено»
В.о. завідувача кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2023 р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Математичні методи криптографічного захисту інформації»

зі спеціальності: 113 Прикладна математика
на тему: **«Усічення цифрового підпису для схем типу Ель-Гамаля»**

Виконав:
студент IV курсу, групи ФІ-94
Кріпака Ілля Анатолійович _____

Керівник:
доц. каф. ММЗІ, к.т.н.
Яковлєв Сергій Володимирович _____

Рецензент:
ст. викладач каф-ри ІБ
Василенко Олексій Дмитрович _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ — 2023

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені Ігоря СІКОРСЬКОГО»
Навчально-науковий фізико-технічний інститут
Кафедра математичних методів захисту інформації

Рівень вищої освіти — перший (бакалаврський)
Спеціальність — 113 Прикладна математика,
ОПП «Математичні методи криптографічного захисту інформації»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Сергій ЯКОВЛЄВ
«__» _____ 2023 р.

ЗАВДАННЯ
на дипломну роботу

Студент: Кріпака Ілля Анатолійович

1. Тема роботи: «*Усічення цифрового підпису для схем типу Ель-Гамаля*», науковий керівник дипломної роботи: доц. каф. ММЗІ, к.т.н. Яковлєв Сергій Володимирович,

затверджені наказом по університету №__ від «__» _____ 2023 р.

2. Термін подання студентом роботи: «__» _____ 2023 р.

3. Об'єкт дослідження: *усічення цифрових підписів типу Ель-Гамаля.*

4. Предмет дослідження: *цифрові підписи типу Ель-Гамаля.*

5. Перелік завдань: *аналіз наявних алгоритмів усічення підписів; побудова алгортиму для усічення цифрового підпису за схемою Ель-Гамаля та ДСТУ 4145-2002; реалізація алгоритму усічення ДСТУ 4145-2002 для перевірки на практиці.*

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: *Презентація доповіді.*

7. Орієнтовний перелік публікацій: *I Міжнародна науково-практична конференеція «Кіберборотьба: розвідка, захист та протидія», XXI Всеукраїнська науково-практична конференція*

«Теоретичні і прикладні проблеми фізики, математики та інформатики»

8. Дата видачі завдання: 10 вересня 2022 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-15 вересня 2022 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	Вересень-жовтень 2022 р.	Виконано
3	Створення методу усічення підпису для ДСТУ 4145-2002	Листопад-грудень 2022 р.	Виконано
4	Створення методу усічення підпису для класичної схеми цифрового підпису типу Ель-Гамаля	Січень-лютий 2023 р.	Виконано
5	Створення методу усічення підпису для узагальненої схеми цифрового підпису типу Ель-Гамаля	Лютий-березень 2023 р.	Виконано
6	Реалізація методу усічення підпису для ДСТУ 4145-2002	Квітень-червень 2023 р.	Виконано
7	Оформлення дипломної роботи	Червень 2023 р.	Виконано

Студент _____ Ілля КРІПАКА

Керівник _____ Сергій ЯКОВЛЄВ

РЕФЕРАТ

Кваліфікаційна робота містить: 52 стор., 18 рисунків, 1 таблицю, 27 джерел.

У даній роботі проводиться огляд та аналіз наявних методів усічення підписів, а також розроблення методів із усічення цифрових підписів для ДСТУ 4145-2002, класичного та узагальнених схем цифрового підпису за Ель-Гамалем. Варто зазначити, що виведенні методи є справедливі і те, що практична перевірка за допомогою програмної реалізації підтвердила можливість із усікання цифрових підписів створених за схемою ДСТУ 4145-2002.

Метою роботи є створення методів для усічення цифрових підписів для схем типу Ель-Гамаля, а саме, для ДСТУ 4145-2002, для класичної та узагальнених схем Ель-Гамаля.

Об'єктом дослідження є усічення цифрових підписів типу Ель-Гамаля.

Предметом дослідження є цифрові підписи типу Ель-Гамаля.

УСІЧЕННЯ ЦИФРОВИХ ПІДПИСІВ, ЦИФРОВІ ПІДПИСИ, ЕЛІПТИЧНІ КРИВІ, СХЕМА ЦИФРОВОГО ПІДПИСУ ЕЛЬ-ГАМАЛЯ, СХЕМА ЦИФРОВОГО ПІДПИСУ ДСТУ-4145-2002

ABSTRACT

The thesis contains : 52 pages, 18 figures, 1 table, 27 sources.

This thesis reviews and analyzes the existing methods of signature truncation, as well as develops methods for truncating digital signatures for DSTU 4145-2002, classical and generalized El-Gamal digital signature schemes. It is worth noting that, the obtained methods by deduction are fair and that practical verification using software implementation has confirmed the possibility of truncating digital signatures, created according to the scheme of DSTU 4145-2002.

The purpose of the thesis is to create methods for the truncation of digital signatures for El-Gamal-type schemes, specifically, for the classical, generalized El-Gamal scheme and DSTU 4145-2002.

The research object of the study is the truncation of the El-Gamal-type digital signatures.

The research subject is El-Gamal-type digital signatures.

TRUNCATING DIGITAL SIGNATURES, DIGITAL SIGNATURES, ELLIPTIC CURVES, EL-GAMAL DIGITAL SIGNATURE SCHEME, DSTU 4145-2002 DIGITAL SIGNATURE SCHEME

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ECC — Elliptic Curve Cryptography

DLP – Discrete Logarithm Problem

ЦП — цифровий підпис

ЕК — еліптична крива

H(M) — результат застосування функції гешування до повідомлення M

LSB — кодування наведене у форматі найменш значущого біта (Least Significant Bit), де найбільші значущі біти знаходяться праворуч, а найменші – ліворуч

HEX — позиційна чистема числення за основою 16 (HEXadecimal), де кожне число записується за допомогою 16 символів

ВСТУП

Актуальність дослідження.

У сучасному світі проблема скорочення підпису має велике значення в легкій криптографії для малоресурсних пристроїв у ситуації, де підписи зберігаються довгий час, але перевіряються відносно рідко. Ми хочемо, щоб підпис скорочувався на стороні підписника без безпосередньої зміни алгоритму, але за рахунок додаткових обчислень на стороні верифікатора. Також очевидною вимогою скороченого алгоритму підпису виступає збереження та незначне зменшення стійкості, оскільки зловмисник може певним чином маніпулювати невідомою частиною підпису.

Метою дослідження є підвищення ефективності реалізацій асиметричних криптосистем для малоресурсних пристроїв. Для досягнення мети необхідно розв'язати задачу дослідження, яка полягає у вдосконаленні методів усічення підписів для схем типу Ель-Гамаля. Для розв'язання задачі необхідно виконати такі завдання:

1) провести огляд опублікованих джерел за тематикою дослідження;

2) провести порівняльний аналіз опублікованих методів усічення цифрових підписів;

3) запропонувати методи усічення підписів для ДСТУ 4145-2002, класичної та узагальненої схем цифрового підпису Ель-Гамаля;

4) практично перевірити запропоновані методи для конкретних реалізацій зазначених криптосистем.

Об'єктом дослідження є інформаційні процеси в системах криптографічного захисту.

Предметом дослідження є схеми цифрових підписів.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: методи лінійної та абстрактної алгебри, дискретної математики, аналізу алгоритмів, методи комп'ютерного моделювання.

Наукова новизна отриманих результатів полягає у тому, що вперше запропоновано метод усічення підпису для національного стандарту ДСТУ 4145-2002, а також для класичної та узагальнених схем типу Ель-Гамаля.

Практичне значення роботи. Одержані результати дозволяють зменшувати кількість необхідної пам'яті для зберігання підписів типу Ель-Гамаля без кардинальної зміни алгоритмів підписування, що має велике значення для малопотужних криптографічних пристроїв.

Апробація результатів та публікації. Результати даної роботи частково представлені на I Міжнародній науково-практичній конференції «Кіберборотьба: розвідка, захист та протидія» (Квітень 20 – 21, м. Київ) та на XXI Всеукраїнській науково-практичній конференції «Теоретичні і прикладні проблеми фізики, математики та інформатики» (Травень 11 – 12, м. Київ).

1 УСІЧЕННЯ ЦИФРОВОГО ПІДПИСУ НА ОСНОВІ СТАНДАРТІВ EDDSA ТА ECDSA

У цьому розділі розглянуто загальні відомості про ЦП та схеми ЦП такі як: Е-Г, DSA, Shnorr. Наведено характеристики еліптичних кривих, що будуть використані у роботі. Проаналізовано особливості схем ЦП на ЕК та відповідні стандарти ECDSA, EdDSA, ДСТУ 4145-2002. Проведено дослідження результатів попередника та його успіхів у скороченні підпису для стандартів ECDSA, EdDSA. Також наведено детальний огляд вищезазначених алгоритмів.

1.1 Загальні відомості про цифрові підписи

Спершу визначимо, що ж таке цифровий підпис. Цифровий підпис — це математична схема для перевірки автентичності, цілісності та визначення приналежності цифрових повідомлень і документів. Цифрові підписи будуються як асиметричні алгоритми. Тобто вони використовуть у собі два ключі:

1) Особитий – для підписування повідомлень/файлів;

2) Відкритий – для перевірки цього підпису іншими користувачами.

Варто зазначити, що сама перевірка відкритим ключем дає велику впевненість у тому, що повідомлення не було скомпрометоване. Саме тому цифрові підписи набули широкого розповсюдження у всіх сферах, починаючи від військового сектору, закінчуючи звичайним обміном повідомлень у мережі.

1.1.1 Цифровий підпис за Ель-Гамалем

Однією із основних схем, що реалізує схему цифрових підписів є схема Ель-Гамаля, яка вперше опублікована та описана у оригінальній статті [7]. Додам, що ця криптосистема побудована на основі складності обчисленнь дискретного логарифма за скінченними полями. Також на основі неї замість множення у кільці $\mathbb{Z}_p$ за простим модулем p можна перевизначити схему на еліптичні криві та інші алгебраїчні структури.

Введемо класичну схему підпису за Ель-Гамалем разом із можливістю підписувати повідомлення довільної довжини. Розгляньмо алгоритм генерації.

Класична схема Ель-Гамаля

Вхідні параметри:

– p – великий простий модуль;

– q – велике просте, що ділить p, тобто $p = q \cdot b$;

– g – елемент порядку q;

– x – особистий ключ $\in_R \mathbb{Z}_p$;

– $y = g^x \pmod p$;

– k – випадково згенероване одноразове число;

– $H : \{0,1\}^* \to \mathbb{Z}_q$ – геш функція.

Відкритий ключ: (y,p,g).

Особистий ключ: (x).

Алгоритм 1.1. [7] Алгоритм генерування та підтвердження підпису за схемою класичного Ель-Гамаля

1) *Підпис:* Для підпису повідомлення m користувач А повинен виконати наступне.

а) Обрати випадкове секретне $k : 1 \leqslant k \leqslant q-1, \gcd(k,q) = 1$.

б) Обчислити $r = g^k \bmod p$.

в) Обчислити $s = k^{-1} \cdot (H(m) - x \cdot r) \bmod q$.

г) (r,s) — підпис.

2) *Перевірка:* Для підтвердження підпису користувача A (r,s) на m, B отримує копії параметрів A (p,g) та відкритого ключа y, користувач B повинен виконати наступне.

а) Перевірити $1 \leqslant r \leqslant p-1$.

б) Перевірити нерівність $y^r \cdot r^s \stackrel{?}{=} g^{H(m)}$.

Алгебраїчні співвідношення, які лежать у основі схеми Ель-Гамаля, можуть бути замінені на інші, що дозволяє генерувати інші схеми із зберіганням основних властивостей. Значну частину таких схем можна описати у формальний спосіб, який одержав назву узагальненої схеми Ель-Гамаля [16]. Ці схеми задаються параметрами A, B, C, що визначаються як певні функції від $H(m)$, s, r.

Узагальнена схема Ель-Гамаля

Вхідні параметри:

– p – великий простий модуль;

– q – велике просте, що ділить p, тобто $p = q \cdot b$;

– g – елемент порядку q;

– x – особистий ключ $\in_R \mathbb{Z}_q$;

– $y = g^x \bmod p$;

– k – випадково згенероване одноразове число, що $\in_R \mathbb{Z}_q^*$;

– A, B, C – співвідношення від m, r, s, що вибираються за співвідношенням $Ak \equiv B + Cx \bmod q$;

– $H : \{0,1\}^* \to \mathbb{Z}_q$ – геш функція.

Відкритий ключ: (y,p,g).

Особистий ключ: (x).

Алгоритм 1.2. [7] Алгоритм генерування та підтвердження підпису за схемою узагальненого Ель-Гамаля

1) *Підпис:* Для підпису повідомлення m користувач А виконує наступне.

а) Обрати випадкове секретне $k : 1 \leqslant k \leqslant q, \gcd(k,q) = 1$.

б) Обчислити $r = g^k \bmod p$.

в) Обчислити $s = k^{-1} \cdot (H(m) - x \cdot r) \bmod q$.

г) (r,s) — підпис.

2) *Перевірка:* Для підтвердження підпису користувача A (r,s) на m, B отримує копії параметрів A (p,g) та відкритого ключа y користувач B повинен виконати наступне.

а) Перевірити $1 \leqslant r \leqslant p-1$.

б) Перевірити нерівність $y^r \cdot r^s \stackrel{?}{=} g^{H(m)}$.

1.1.2 Цифровий підпис Шнорра

ЦП за схемою вперше було опубліковано 1991 року [17] та запропоновано Клаусом Шнорром. Вона використовує перетворення, які базовані на проблемі дискретного логарифмування (DLP) разом із геш функцією, що є одним із найважливіших плюсів схеми – ефективність обчислення в порівнянні із іншими схемами. Варто зазначити, що головною відмінністю від схеми Ель-Гамаля є вперше використана геш функція для побудови підпису. Загалом схема підпису за схемою Шнорра була перспективною альтернативою іншим схемам, але не набула широкого вжитку завдяки патенту, що не дозволяв усім без його купівлі використовувати у своїх системах. Додам, що лише тільки у 2008 році, разом із закінченням терміну дії патенту, громадськість отримала доступ до нього. Розгляньмо наступну схему [6]

Вхідні параметри:

– p – великий простий модуль;

– g – елемент порядку $p-1$;

– x – особистий ключ $\in_R \mathbb{Z}_p$;

– $y = g^x \bmod p$;

– k – випадково згенероване одноразове число;

– $H : \{0,1\}^* \to \mathbb{Z}_q$ – геш функція.

Відкритий ключ: (y,p,g). **Особистий ключ:** (x).

Алгоритм 1.3. [13] Підпису та підтвердження підпису за схемою

Schnorr

1) *Підпис:* Для підпису повідомлення m користувач A робить наступне.

а) Обрати випадкове секретне $k : 1 \leqslant k \leqslant q - 1$.

б) Обчислити $r = g^k \bmod p$.

в) Обчислити $e = H(r||m)$, де $||$ – конкатенація.

г) Обчислити $s = k + x \cdot e \pmod q$.

(e,s) — підпис.

2) *Перевірка:* Для підтвердження підпису користувача A (e,s) на m, B отримує копії параметрів A (p,q,g) та відкритого ключа y користувач B повинен виконати наступне.

а) Обчислити $r^{'} = g^s y^{-e} \bmod p$.

б) Обчислити $e = H(r||m)$, де $||$ – конкатенація.

в) Перевірити нерівність $e \stackrel{?}{=} H(r^{'}||m)$.

1.1.3 Цифровий підпис DSA

На противагу схемі Ель-Гамаля, NIST запропонувало свій варіант стандарту у 1991 році, а в 1994 було адаптовано та опубліковано у FIPS-186 [19]. Варто сказати, що DSA є певним збірним варіантом підписів Е-Г, Шнорра. Цей алгоритм підпису так само базується на складності обчислення DLP, але має дещо інші перетворення для створення підпису та його перевірки. На даний час актуальна уже 4 версія стандарту [8], в якій можна побачити, що обмеження на геш функцію SHA-1 уже немає. Можна обрати різні довжини повідомлень і під нього підлаштувати відповідну геш функцію. То ж розглянемо алгоритм генерації.

Вхідні параметри:

– p – великий простий модуль;

– q – простий дільник $(p - 1)$, тобто $q|(p - 1)$;

– x – особистий ключ $\in_R \mathbb{Z}_p$;

– $y = g^x \bmod p$;

– k – випадкове секретне одноразове число;

– $H : \{0,1\}^* \to \mathbb{Z}_q$.

Відкритий ключ: (p,q,g).

Особистий ключ: (x).

Алгоритм 1.4. [9] Алгоритм підпису та підтвердження підпису за схемою DSA

1) *Підпис:* Для підпису повідомлення m користувач А повинен виконати наступне.

а) Обрати випадкове секретне $k : 1 \leqslant k \leqslant q-1, gcd(k,q) = 1$.

б) Обчислити $X = g^k \bmod p$, $r = X \bmod q$.

в) Обчислити $k^{-1} \pmod q$.

г) Обчислити $e = H(m)$.

д) Обчислити $s = k^{-1}(e + x \cdot r) \pmod q$. Якщо $s = 0$, то повернутися на крок 1.

е) (r,s) — підпис.

2) *Перевірка:* Для підтвердження підпису користувача A (r,s) на m, B отримує копії параметрів A (p,q,g) та відкритого ключа y, користувач B повинен виконати наступне.

а) Перевірити чи r, s входять у інтервал $[1, q-1]$.

б) Обчислити $e = H(m)$.

в) Обчислити $w = s^{-1} \bmod q$.

г) Обчислити $u_1 = e \cdot w \pmod q$ та $u_2 = r \cdot w \pmod q$.

д) Перевірити нерівність $((g^{u_1} \cdot y^{u_2}) \bmod p) \bmod q \stackrel{?}{=} r$.

1.2 Еліптичні криві та схеми цифрового підпису

Спершу визначимо, що таке еліптична крива. Еліптичною кривою називається множина точок, що задовольняє рівняння разом із фіктивною точкою, що називається точкою на нескінченності і

позначається O_E. Також вона позначається як $E/E/F$ та задається рівнянням від двох змінних x та y. Пара елементів (x,y), де x,$y \in F$ – називається точкою.

Царина, де розвивається еліптична криптографія зветься – ECC (Elliptic Curve Cryptography). Вона має широке використання у цифрових підписах (наприклад: ECDSA, EdDSA, і т.д.). Головним плюсом ECC над загальновживаними схемами ЦП є здатність забезпечити той же рівень безпеки разом із меншими за розміром ключами. Цей факт і робить їх привабливими для преходу, де обчислювальні потужності пристроїв є обмеженими, але разом із розвитком електроніки та квантових комп'ютерів, всі поступово почали задумуватися про перехід до пост-квантових стандартів. Так, алгоритм дискретного логарифмування Шора [18] ще не досить ефективно реалізований, але над його покращенням активно працюють.

Тепер, переходячи до огляду еліптичних кривих, треба визначити, що таке порядок у еліптичної кривої. Порядком кривої будемо називати кількість точок, що у ній міститься. У більшості ЕК порядок не є простим, що із одного боку є погано, але із іншого вони все ще забезпечують потрібну стійкість. Припустимо, що порядок кривої $= h * n$, де h – кофактор, n– порядок простої підгрупи кривої.

Для наочності розгляньмо криві Веєрштрасса.

1.2.1 Криві Веєрштрасса

Криві Ваєштрасса мають багато рівнянь для різних характеристик та ми розглянемо двоє із них:

1) над простим полем $\mathbb{F}_q$, де $\operatorname{char}\mathbb{F} \neq 2{,}3$;

2) над полем $\mathbb{F}_{2^m}$, де $\operatorname{char}\mathbb{F} = 2$.

Зауваження. ЕК є *гладкою* тоді і тільки тоді, коли крива без заламів та самоперетинів.

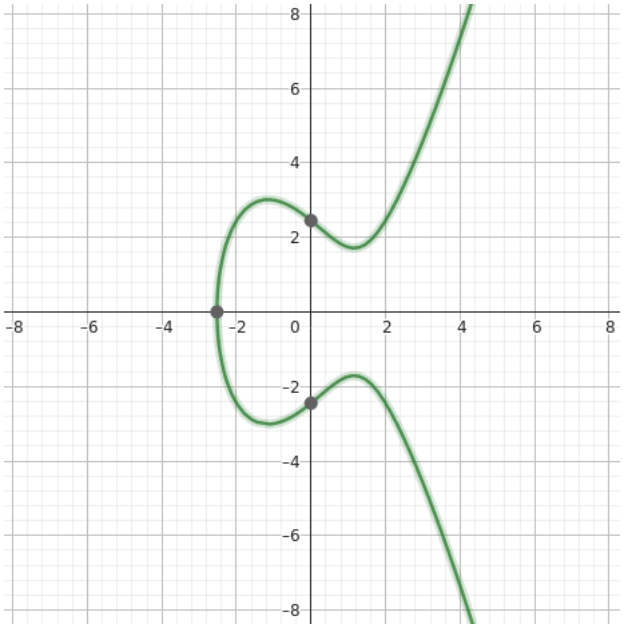

Рисунок 1.1 – Приклад кривої Веєштрасса $y^2 = x^3 - 4x + 6$.

1) Криві над полем виду $\mathbb{F}_q$, де $char\mathbb{F} \neq 2{,}3$, q – просте
Криві цього виду мають наступну форму:

$$W_{a,b} : y^2 = x^3 + ax + b \tag{1.1}$$

Ці криві є гладкими за умови виконання умови гладкості:

$$4a^3 + 27b^2 \neq 0.$$

Варто зазначити, що криві Веєштрасса є симетричними відносно x. Також вона має такий порядок, за наступних умов:

а) для псевдовипадкових кривих, $h = 1$ та n – просте;

б) для специфічних кривих, $h > 1$ та n – не є простим.

Арифметика задається наступним чином [5]:

– Для кожної точки $P = (x,y)$ на кривій Веєштрасса $W_{a,b}$ точка на нескінченності O_E служить як нейтральна точка, тобто $P + O_E = O_E + P = P$.

– Для кожної точки $P = (x,y)$ на кривій $W_{a,b}$, $-P$ це точка $(x, -y)$ і вона має властивість $P + (-P) = O_E$.

– Припустимо, ми маємо $P_1 = (x_1,y_1), P_2 = (x_2,y_2)$ на кривій $W_{a,b}$, де

$P_1 \neq \pm P_2$ і нехай $Q = P_1 + P_2 = (x,y)$, тоді будемо мати наступні формули для обчислення координат точок:

$$x = \left(\frac{y_1 - y_2}{x_1 - x_2}\right)^2 - x_1 - x_2, \tag{1.2}$$

$$y = -y_1 + \frac{y_1 - y_2}{x_1 - x_2}(x_1 - x). \tag{1.3}$$

– Припустимо, ми маємо $P_1 = (x_1,y_1)$ на кривій $W_{a,b}$, де $P \neq -P$ і нехай $Q = 2P$. Тоді $Q = (x,y)$, де

$$x = \frac{3x_1^2 + a}{2y_1}^2 - 2x_1 \tag{1.4}$$

$$y = \frac{3x_1^2 + a}{2y_1}(x_1 - x) - y_1 \tag{1.5}$$

2) Криві над полем виду $\mathbb{F}_{2^m}$, де $\operatorname{char}\mathbb{F} = 2$. Криві цього виду також називають у стандартах «Binary Curves» мають наступну форму:

$$B_{a,b} : y^2 + xy = x^3 + ax^2 + b, \tag{1.6}$$

де ця крива є гладкою за умови $b \neq 0$.

За умови вибору псевдовипадкових параметрів крива має $h = 2$.

Арифметика задається наступним чином [5]:

– Для кожної точки $P = (x,y)$ на кривій Веєштрасса $B_{a,b}$ точка на нескінченності O_E служить як нейтральна точка, тобто $P + O_E = O_E + P = P$.

– Для кожної точки $P = (x,y)$ на кривій $B_{a,b}$, $-P$ це точка $(x,x + y)$ і вона має властивість $P + (-P) = O_E$.

– Припустимо ми маємо $P_1 = (x_1,y_1)$,$P_2 = (x_2,y_2)$ на кривій $B_{a,b}$, де $P_1 \neq \pm P_2$ і нехай $Q = P_1 + P_2 = (x,y)$. Тоді формули додавання будуть

мати форму:

$$x = \frac{y_1 + y_2}{x_1 + x_2}^2 + \frac{y_1 + y_2}{x_1 + x_2} + a - x_1 - x_2, \tag{1.7}$$

$$y = \frac{y_1 + y_2}{x_1 + x_2}(x_1 + x) - y_1 - x. \tag{1.8}$$

– Припустимо ми маємо $P_1 = (x_1,y_1)$ на кривій $B_{a,b}$, де $P \neq -P$ і нехай $Q = 2P$. Тоді $Q = (x,y)$, де обчислення x та y виконується відносно таких формул:

$$x = (x_1 + \frac{y_1}{x_1})^2 + x_1 + \frac{y_1}{x_1} + a = x_1^2 + \frac{b}{x_1^2}, \tag{1.9}$$

$$y = (x_1 + \frac{y_1}{x_1})(x_1 + x) - y_1 - x. \tag{1.10}$$

1.2.2 Закручені криві Едвардса

Криві Едвардса — це сімейство еліптичних кривих, що були досліджувані Гарольдом Едвардсом у 2007 році. А от саме закручені криві були представлені Даніелем Бернштейном у 2008 році [5] та вперше застосувані для криптографії, зокрема у стандарті підпису EdDSA. Завдяки значному прирості у обчисленнях по відношенню до кривих Веєрштрасса вони є підходящими для малоресурсних IoT (Internet of Things) пристроїв [14]. Розгляньмо криву Ed25519.

Ця крива має наступну форму:

$$E_{a,d} : ax^2 + y^2 = 1 + dx^2y^2$$

Порядок кривої не є простим, тобто $h = 8$, n – просте. Ще один важливий факт, закручені криві Едвардса є симетричними відносно y. Арифметика тут задається наступним чином [5]:

– Припустимо, ми маємо $P_1 = (x_1,y_1)$, $P_2 = (x_2,y_2)$ і, нехай,

$Q = P_1 + P_2 = (x,y)$, тоді додавання буде виглядати так:

$$(x_1,y_1) + (x_2,y_2) = \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - ax_1x_2}{1 - dx_1x_2y_1y_2} \right) = (x,y).$$

– Ця ж формула працює і для подвоєння $Q = 2P$, що має вигляд:

$$2P = (x_1,y_1) + (x_1,y_1) = \left(\frac{2x_1y_1}{1 + dx_1^2y_1^2}, \frac{y_1^2 - ax_1^2}{1 - dx_1^2y_1^2} \right) = (x,y).$$

– Нейтральний елемент є точкою (0,1). Припустимо, що $P = (x,y)$ тож для кожної точки на закрученій кривій Едвардса $P + (0,1) = (x, y) = P$.

– Обернений елемент до точки $P = (x,y)$ — $-P = (-x,y)$.

– Також до попереднього пункту $P + -P = O_E$.

– $\mathrm{ord}(0, -1) = 2$.

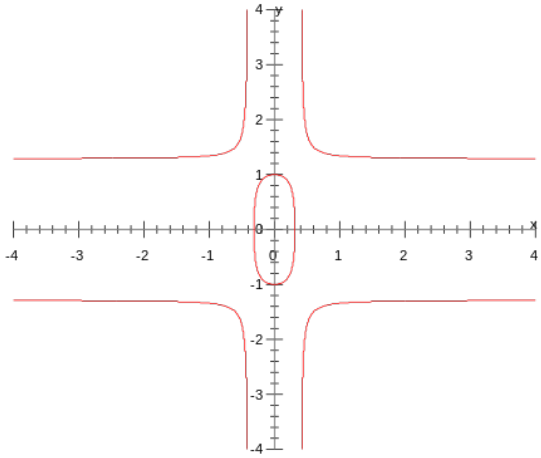

Рисунок 1.2 – Приклад закрученої кривої Едвардса $10x^2 + y^2 = 1 + x^2y^2$.

1.2.3 Цифровий підпис ECDSA

ECDSA розшифровується як Elliptic Curve Signature Algorithm, стандарт випущений NSA, що був вперше опублікований у 1999 році [9], широко використовуваний алгоритм для цифрових підписів. Ця схема ЦП

є певною компіляцією двох стандартів DSA, Schnorr, але уже із використанням ЕК. В загальному вважається досить стійким до атак, але до NSA і випущених стандартів є певна недовіра пов'язана із специфічними еліптичними кривими та із DRBG(Deterministic Random Bit Generator), над якими працює агенція та випускає у стандарти [10]. Але все ж таки розгляньмо даний стандарт.

Параметри ECDSA:

– q – розмір поля, що лежить у основі;

– a – параметр ЕК (рівний $q-3$ у P-256);

– b – параметр ЕК;

– $G=(x_G,y_G)$ – базова точка;

– n – порядок базової точки G;

– H – будь-яка схвалена геш функція.

Деталі із генерації параметрів наявні за посиланням [2].

Алгоритм 1.5. Алгоритим підписування та підтвердження підпису за схемою ECDSA [2]

1) *Підпис:* Для підпису повідомлення m користувач A повинен виконати наступне.

а) Згенерувати (k,k^{-1}), де k – секретне число та k^{-1} – секретне число по $\pmod n$.

б) Обчислити точку на кривій $R=kG=(x_r,y_r)$.

в) Обчислити $e=H(m)$ (вважаємо що вихід подається у бітовому рядку).

г) Обчислити $s=(k^{-1}\cdot(e+d\cdot r))(\pmod n)$.

д) (r,s) — підпис.

2) *Перевірка:* Для підтвердження підпису користувача A (r,s) на m, B отримує копії параметрів A (q,a,b,G,n) та відкритого ключа Q, користувачу B потрібно виконати наступне.

а) Перевірити чи $r,s\in[1,n-1]$.

б) Обчислити $e=H(m)$.

в) Обчислити $w=s^{-1} \bmod n$.

г) Обчислити $u_1 = (e \cdot w) \bmod n$, $u_2 = (r \cdot w) \bmod n$.

д) Обчислити точку на еліптичній кривій $R = (x_r, y_r) = u_1 G + u_2 Q$.

е) Обчислити $v = x_r \bmod n$.

ж) Перевірити нерівність $v \stackrel{?}{=} r$.

1.2.4 Цифровий підпис EdDSA

EdDSA розшифровується як Edvards-curve Digital Signature algotithm, побудований на основі схем підпису Schnorr, DSA та ECDSA лише на закручених еліптичних кривих Едвардса. Сам алгоритм був опублікований 2011 року [4] і, зокрема, заточений на швидкість і безпеку. На даний час він є доволі новим стандартом, на який активно переходять користувачі.

Параметри EdDSA:

– $b = 256$ для Ed25519;

– Запропонована міра безпеки (характеризує бітову довжину параметрів): 128 біт для Ed25519, позначимо – $BS(BitStrength)$;

– $G = (x_G, y_G)$ – базова точка;

– $c = 3$ для Ed25519;

– s – виходить із переведення першої половини $H(d)$ в кодування числа;

– $Q = sG$;

– H – рекомендовані до використання геш функції (SHA512, $\cdots$ і т.д.).

Алгоритм 1.6. Алгоритм підписування та підтвердження підпису за схемою EdDSA [20]

1) *Підпис:* Для підпису повідомлення m користувач A повинен виконати наступні кроки.

а) Обчислити $H(d) = (\texttt{hdigest}_1 || \texttt{hdigest}_2)$, використовуючи дану геш функцію.

б) Використовуючи другу половину попереднього гешу, порахувати $R = H(\texttt{hdigest}_2||m)$.

в) Обчислити точку rG.

г) $S = (r + H(r||Q||m) \cdot s) \bmod n$

д) $(R||S)$ — підпис

2) *Перевірка:* Для підтвердження підпису користувача A $(R||S)$ на m, B отримує копії параметрів A (b,BS,G,c,s) та відкритого ключа Q, користувачу B потрібно виконати наступне.

а) Сформувати бітовий рядок $HashData = R \parallel Q \parallel M$.

б) Обчислити $H(HashData)$ та інтерпретувати рядок як число t.

в) Перевірити нерівність $(S)G \stackrel{?}{=} R + tQ$.

1.2.5 ДСТУ 4145-2002

Розгляньмо тепер національний стандарт – ДСТУ 4145-2002. ДСТУ 4145-2002 було пирийнято у 2002 році. І лише у 2022 році повнітю було замінено попередній ще радянський стандарт ДСТУ ГОСТ 28147:2009 [26]. Сама схема активно використовується у всіх сферах надання цифрових послуг [23]. Також був досліджений попередниками на винайдення швидшого методу обчислення [27] та на приховані канали передачі [22]. Зауважимо, що ДСТУ 4145-2002 у собі використовує еліптичні криві над полем $\mathbb{F}_{2^m}$ (1.6).

Параметри ДСТУ 4145-2002 [25, 22] :

– m – степінь розширення поля $\mathbb{F}_2$ вибирається випадково із таблиць стандарту;

– $P = (x_P, y_P)$ – базова точка;

– d – особистий ключ (обчислюється за допомогою генертатора псевдовипадкових чисел);

– $Q = -dP$;

– n – порядок базової точки, просте непарне;

– T – вхідне повідомлення довжини $L_T > 0$;

– L_D – довжина цифрового підпису;

– H – ГОСТ 34.311 або будь-яка інша функція гешування, рекомендована уповноваженим органом державної влади у сфері криптографічного захисту інформації.

Алгоритм 1.7. Алгоритм підписування та підтвердження підпису за схемою ДСТУ 4145-2002 [20]

1) **Обчисленння передпідпису:** Для обчислення підпису для повідомлення коритувачу спочатку треба обчислити передпідпис, що використовує загальні параметри цифрового підпису. Користувач А повинен зробити наступне.

а) Обчислити випадкове ціле число e.

б) Обчислити точку еліптичної кривої $R = eP = (x_R, y_R)$.

в) Якщо $x_R = 0$, перейти на крок 1), як ні, то присвоїти $F_e = x_R$.

У результаті маємо сформований передпідпис F_e, що відповідає таємному випадкому параметру e, де e – ціле число, $0 < e < n$, $F_e \in \mathbb{F}_{2^n}$.

2) *Підпис:* Для підпису повідомлення m користувач А повинен виконати наступне.

а) Перевірити на правильність параметри ЕК, загальні параметри ЦП, особитий ключ, кратність 16 числа L_D, ідентифікатор геш-функції та нормативні документи на обмеження по довжині повідомлення.

б) Обчислити $e = H(T)$.

в) Перетворити результат гешування e на елемент цифрового поля h.

г) Обираємо передпідпис F_e із готової таблиці разом із значенням e або обчислюємо за алгоритмом вище.

д) Обчислити елемент основного поля $y = hF_e$.

е) Перетворити елемент поля y на ціле число r.

ж) Обчислити ціле число $s = (e + dr) \bmod n$.

Повернути підпис D, що є певним перетворенням пари (r,s) на ЦП довжини L_D.

3) *Перевірка:* Для підтвердження підпису користувача A D на m, B

отримує копії загальних параметрів ЦП A та відкритого ключа Q. Користувач B повинен виконати наступне.

а) Перевірити на правильність ідентифікатор геш-функції, кратність 16 числа L_D(отримано від геш функції), загальні параметри ЦП, відкритий ключ.

б) Обчислити за повідомленням m, $H(m) = e$.

в) Перетворити геш-код e на елемент цифрового поля h.

г) Перетворити ЦП D на пару (r,s).

д) Обчислити точку еліптичної кривої $R = sP + rQ, R = (x_R, y_R)$.

е) Обчислити елемент основного поля $y = hx_R$.

ж) Перетворити елемент y основного поля на ціле число $r^{'}$.

з) Якщо $r = r^{'}$, то повернути що підпис дійний, інакше – не дійсний.

1.3 Задача зменшення розміру підпису

Надалі проведемо аналіз результатів Томаса Проніна, що були викладені у статті [15]. Загалом задачу зменшення підпису можна вирішити у різні способи, зокрема, можна застосувати:

– менші еліптичні криві, до прикладу, NIST P-192, що досі дає потрібний рівень безпеки у 96 біт та видає на вихід підписи із 48 байт;

– модифікації з алгоритмом підпису на основі схеми Schnorr на EK. Позаяк ми зупинилися на стандартах EdDSA на Ed25519 та ECDSA на P-256, значить варіанти написані вище не можуть бути використані. Тут ми розглянемо методи, якими можна скорочувати підписи.

В загальному, обрізання самого підпису може бути застосоване для будь-яких схем підпису. Тобто будемо упускати останні t бітів підпису, відповідно для верифікатора треба буде перебрати 2^t можливих варіантів закінчення підпису. В середньому це передбачає виконання перевірки за 2^{t-1} ітерацій. Саме цей процес такого собі «урізання підпису» можна назвати безпечним завдяки тому, що:

– обрізання підпису і реконструкція використовує лише відкриті дані;

– справжній необрізаний підпис отримуєтья як побічний продукт від перебору.

Іншими словами, якщо підробка можлива для усіченого підпису, то зловмисник повинен так само запусти перевірку для підробленого підпису, щоб і повний підпис теж був підробленим. Але це не полегшує роботу зловмиснику. Спроби із бенчмарками на різних схемах та довжинах буде доступне за покликанням:

https://github.com/pornin/crrl/.

Зауваження. Зазначу, що у даній реалізації будемо використовувати усічення підпису від 8 до 32 біт.

Перед розглядом алгоритмів усічення спершу варто ввести алоритм BSGS(Big Steps Giant Steps) для задачі пошуку дискретного логарифму.

Алгоритм 1.8. Великих та Малих Кроків (BSGS) [12]

Вхід:

– α – генератор групи G;

– $\beta \in G$;

– n – порядок груи G.

Вихід: $x = \log_\alpha \beta$.

1) $m = [\sqrt{n}] + 1$.

2) $b = \alpha^m$.

3) Будуємо таблиці:

– $\{b^u : u = \overline{1,m}\}$ – кроки велетня.

– $\{\beta \cdot \alpha^v : v = \overline{1,m}\}$ – кроки дитини.

4) Знаходимо в таблицях однакові елемети b^{u^*} та $\beta \cdot \alpha^{v^*}$.

5) Обчислюємо $x = m \cdot u^* - v^* \pmod n$.

1) **Підписи EdDSA**

В описі будемо використовувати наступну нотацію:

– Відкритий ключ це точка на кривій – Q.

– Генератор кривої – G.

– Підпис складається із точки на кривій R та скаляру s, де він береться по модулю підгрупи порядку $l \approx 2^{252}$.

– Алгоритм перевірки потребує обчислення значення k, що є певним скаляром. Він виходить у результаті обчислення гешу до конкатенованої точки R, як наведено в алгоритмі, детальніше у розділі 1.6. Це число буде лежати у проміжку $[0, l-1]$. Відповідно до схеми верифікації, вважаємо, що підпис є правильним за виконання умови:

$$8sG = 8R + 8kQ. \tag{1.11}$$

Зауваження. Тут ми будемо використовувати кофактор, позаяк це певний механізм спростити верифікацію і зробити її несуперечною у деяких випадках. Кофактор має властивість такого собі «відображення» точки кривої у підгрупу простого порядку [21]. Для кривої Ed25519 кофактор буде рівним 8 (1.2.2), тому множення точки на це число нічого не змінить, лише зменшить кількість можливих помилок при перевірках.

Основний алгоритм

Зауваження. Надалі усі наші перетворення будуть стосуватися відновлення і певного укорочення s.

Позаяк s (знаходиться у $[0, l-1]$) урізання числа еквіваленте взяттю за модулю 2^n, де $n = 256 - t$. Припускаємо, що $t \geqslant 8, t \in$ [8,32] (1.3). Зауважимо, що перші 3 біти будуть нулями, тобто із цього випливає, що $l < 2^{253}$.

Чому 3 нулі?

Маємо нерівність, що множиться на кофактор, тобто отримуємо елемент у простій підгрупі. Знаючи, що кофактор у даній схемі є рівний 8, то можна сказати, що при кодуванні точки, щоб дізнатися, якій підгрупі вона належить, треба використати на початку 3 біти інформації. В результаті множення на кофактор і відображення точки

на просту підгрупу отримуємо, що перші 3 біти повинні бути рівні нулю.

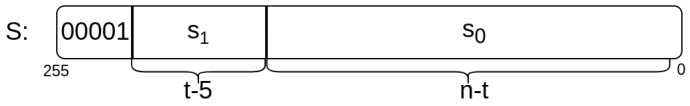

Рисунок 1.3 – Формат s проілюстрований на діаграмі.

Нехай $s_0 = s \pmod{2^n}$, тоді отримаємо, що

$$s = s_0 + 2^{251} + s_1 2^n. \tag{1.12}$$

Отримуємо, що $-2^{(t-5)} \leqslant s_1 \leqslant +2^{(t-5)}$ (значення $+2^{(t-5)}$ можливе у випадку $s \geqslant 2^{252}$).

Чому саме такі межі для s_1?

Відповідно до рисунка 1.12 на початку числа виникає ситуація, де маємо із 255 по 253 біти нулі. 251 біт є обов'язково одиницею, але залишається один 252 біт, у якому не можемо визначити значення. Позаяк числа можуть бути дещо більшими за 2^{251}, то значення цього біта буде мінятися відповідно до знаку.

Тепер підставимо (1.12) у (1.11):

$$s_1(2^n 8G) = 8R + 8kQ - 8(s_0 + 2^{251})G. \tag{1.13}$$

Покладемо у $U = 2^{n+3}$ та $V = 8(R + kQ - 8(s_0 + 2^{251})G$. Тепер обрізання підпису стає проблемою знаходження розв'язку s_1 у межах $[-2^{t-5}, +2^{t-5}]$:

$$s_1 U_1 = V. \tag{1.14}$$

Надалі треба пригадати, що точки $P, -P$ у закрученій кривій Едвардса мають однакові координати y. Зауважимо, що сам алгоритм ідентичний до BSGS (1.8), але має дещо іншу форму.

Алгоритм 1.9. Алгоритм знаходження відповідного s_1 для відновлення

підпису за схемою EdDSA

а) Нехай I та J – це два додатних числа такі, що $IJ \geqslant 2^{(t-5)}$. На початку вважаємо що I,$J = 2^{(t-5)/2}$.

б) Для $j = 0$ до J, визначимо $U_j = jIU$. Будемо обчислювати та накопичувати координати y від точок U_j.

в) Для $i = 0$ до $I-1$, визначимо $V_i = V - iU$. Будемо обчислювати точки V_j та витягувати із них точки y.

г) Дивимося на збіг координат y точок U_j та V_j. Для будь-яких збігів маємо двох кандидатів на розв'язок: $s_1 = i + Ij$ та $s_1 = i - Ij$. Кожного кандидата верифікуємо за нескороченим виразом (1.11).

2) **Підписи ECDSA**

Тепер розглянемо підпис за схемою ECDSA. Сам процес урізання багато в чому нагадує попередній алгоритм для EdDSA. Цей стандарт працює на кривій P-256 (крива Вейерштрасса за $\mathbb{F}_q$, тобто крива не має кофактора, тож необхідності домножувати нерівність на коефіцієнт теж немає). Також вважаємо, що підписи будуть в межах 64 байт. В описі будемо використовувати наступну нотацію:

– Відкритий ключ це точка на кривій – Q;

– Генератор кривої – G;

– Порядок групи є $l \approx 2^{256}$;

– Підпис складається із пари чисел (r,s), що подаються по модулю l;

– Повідомлення m повинне бути пропущене через геш функцію для верифікації та ми так і будемо писати m для спрощення.

Основною нерівністю для перевірки підпису за стандартом описаним вище (1.5) буде:

$$x_r = r \bmod l, R = (m \cdot s^{-1})G + (r \cdot s^{-1})Q.$$

Перепишімо у зручній формі:

$$sR = mG + rQ,\ x_r = r \bmod l \tag{1.15}$$

Зауваження. Тут вважаємо, що точка $R = (x_r,y_r)$.

До алгоритму перевірки усічення треба додати певну оптимізацію, щоб ми були впевнені, що s, яке прийшло на вхід, точно було $\leqslant 2^{255}$. Воно полягає у тому, що, якщо на вхід приходить $s \geqslant 2^{255}$, то треба замінити попердене s на $l - s$. Це виходить із факту, що, якщо (r,s) правильний підпис, то і $(r, -s)$ теж правильний із точністю до навпаки. Таке може утворитися за допомогою властивостей кривої Веєштрасса(P-256), що ми тут використовуємо. А саме, точки $R, -R$ мають однакові координати x. Така собі «заміна» у алгоритмі буде врахована.

Основний алгоритм

Так як ми застосовували певну оптимізацію над s, використаймо її. Знаємо, що тепер $s < 2^{255}$, тобто це можна записати у форматі: $s = s_0 + s_1 2^n$, де $n = 256 - t, t \in [8,32], 0 \leqslant s_1 < 2^{t-1}$.

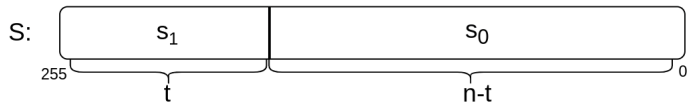

Рисунок 1.4 – Формат s проілюстрований на діаграмі.

Підставляємо виражене s у попередню нерівність (1.11):

$$s_0 R = s_1(2^n R) = mG + rQ, x_r = r \bmod l. \tag{1.16}$$

Алгоритм 1.10. Алгоритм знаходження відповідного s_1 для відновлення підпису за схемою ECDSA

а) Нехай I та J є додатними числами такі, що $IJ \geqslant 2^{t-1}$. Так як і раніше $J = 2^{(t-2)/2}$ та $I = 2J$.

б) Для $j = 0$ до J, визначимо $U_j = s_o R + jI(2^n R)$. Обчислюємо та накопичуємо координати x від точок U_j.

в) Для $i = 0$ до I, обчислюємо $V_i = mG + rQ - i(2^n R)$ та перевіряємо чи координата x співпадає із одною у U_j. Якщо є співпадіння, то ми маємо 2 кандидати: $s = s_0 + (i + Ij)2^n$ та $s = s_0 + (-i + Ij)2^n$.

Якщо розв'язок s_1 існує, то ми спершу дивимося чи змінювали на початку

алгоритму s за допомогою оптимізації. Якщо так, то преходимо до кроку а), як ні, то до б).

а) Позаяк маємо розв'язок s_1, то $s_1 = a + Ib$, де $0 \leqslant a < I$ та $0 \leqslant b < J$. Тоді можна вивести наступне рівняння для перевірки, що веде до того, що $U_b = V_a$, $i = a$, $j = b$ та $s = s_0 + (i + Ij)2^n$.

$$s_0R + (a + Ib)(2^nR) = mG + rQ, x_r = r \bmod l.$$

б) В іншому випадку виходить, що ми будуємо нерівність для $-R$, замість R. Тож нерівність стає такою:

$$\begin{aligned} -(s_0R + (a + Ib)(2^nR) &= mG + rQ \Rightarrow \\ \Rightarrow -(s_0R + I(b + 1)(2^nR)) &= mG + rQ - (I - a)(2^nR) \end{aligned} \tag{1.17}$$

Тобто $-U_{b+1} = V_{1-a}$, із цього отримаємо збіг серед координат із U_{b+1} та V_{1-a}. Маємо $i = I - a$ та $j = b + 1$, отже розв'язком буде $s = s_0 + (I - i + I(j - 1))2^n = s_0 + (-i + Ij)2^n$.
Сам процес верифікації зупиняється як тільки ми знаходимо розв'язок.

Висновки до розділу 1

У розділі розглянуто основні алгоритми засновані на дискретному логарифмуванні, загальні характеристики ЕК та стандарти побудовані на них: ECDSA, EdDSA, ДСТУ 4145-2002.

Крім того, було проаналізовано спосіб, яким було зроблено усічення підпису для стандартів ECDSA та EdDSA. Наведено схематичний алгоритм та пояснення до нього.

2 РОЗРОБКА УСІЧЕННЯ ПІДПИСУ ДЛЯ ІНШИХ АЛГОРИТМІВ

Цей розділ присвячено результатам дослідження можливого скорочення підписів, виведенню алгоритмів для перевірки усіченого підпису за алгоритмами постановки ДСТУ 4145-2002 та Ель-Гамаля.

2.1 Можливі методи усічення підпису

У роботі [3] було розглянуто три можливих методи скорочення підписів. Саме за допомогою них більшість дослідників реалізовують схеми зменшення підпису, але, варто зазначити, що їх кількість може бути більша. Нижче буде наведено їхню ідею та стилий опис.

1) **Схема із використанням функції компресії**

Перед описом схеми варто зазначити, що довжина підпису Ель-Гамаля складає $\lceil \log_2 q \rceil + \lceil \log_2 p \rceil$ бітів. До прикладу, якщо обрати p як число із 1024 бітами, а q із 160 бітами, то довжина підпису буде $1024 + 160 = 1184$ бітів.

У цій схемі пропонується робити скорочення за рахунок деякої функції компресії. Найбільш вдала реалізація цього методу є у стандарті цифрового підпису DSA [8], де значення r, s обчислюються за формулами із алгоритму 1.4:

$$r = (g^k \bmod p) \bmod q,$$
$$s = (k^{-1}(H(m) + xr)) \bmod q,$$

а перевірочне співвідношення має такий вид:

$$r \stackrel{?}{\equiv} ((g^{H(m)s^{-1} \bmod q} \cdot y^{rs^{-1} \bmod q}) \bmod p) \bmod q.$$

Перевагою цього методу є його довжина підпису, яка складає $\lceil \log_2 q \rceil + \lceil \log_2 q \rceil$ бітів. До прикладу, в класичній схемі Ель-Гамаля довжина складає 1184 біт, а от у DSA уже буде $\lceil \log_2 (r \cdot s) \rceil = 160 + 160 = 320$ бітів. Варто зазначити, що застосування цього методу до схем типу Ель-Гамаля вимагає їх значної алгоритмічної перебудови, фактично – побудови нової схеми цифрового підпису.

2) **Фіксування частини підпису**

Як відомо, підписом є пара чисел (r, s). Тоді для зменшення підпису зафіксуємо певне значення як `const` і будемо шукати r як:

$$r = r_0 \parallel \texttt{const},$$

де r_0 – відома частина підпису. Щоб отримати такий підпис, будемо перебирати можливі комбінації із випадкових векторів, що додаються до повідомлення. Сам підпис буде мати форму (r_0, s). Таким чином, не зменшуючи рівень безпеки, будемо приймати за `const` певне фіксоване значення, яке буде відомо заздалегідь обом сторонам. Відмітимо, що:

– цей підхід вимагає збільшення ресурсних витрат у сторони, яка підписує, і тому для малопотужних пристроїв цей метод слабко застосовний;

– цю схему можна використовувати як для r, так і для s; підпис матиме форму (r_0, s) або (r, s_0) відповідно.

3) **Відкидання частини підпису**

Основна ідея методу полягає в обрізанні частини s із підпису. s має наступну форму:

$$s = s_0 \parallel s_1.$$

Сам підпис буде мати наступну форму: (r, s_0). Цей спосіб уже був розглянутий у [3], але Томас Порнін [15] незалежно дослідив даний метод для схем цифрового підпису ECDSA/EdDSA та запропонував його ефективну реалізацію. Аналогічний метод для стандарту ДСТУ 4145-2002 було запропоновано у [11] та для класичної, узагальненої схеми

Ель-Гамаля у [24].

Щодо використання методів у даній роботі, то методи №1 та №2 не підходять із наступних причин:

1) Даний метод вимагає кардинальної перебудови алгоритму. Погляньмо на ті ж схеми Ель-Гамаля та DSA. Що перша, що друга схеми використовують модулярну арифметику, але вони кардинально відрізняються за схемою перевірки та генерації підпису. Позаяк DSA сформульована на таке собі «зменшення» підпису на противагу до Ель-Гамаля.

2) Цей метод використовує повторну генерацію підпису для усічення, що дає відчутно більший час для генерації підпису. В середньому прийдеться чекати $\log_2 \sqrt{\texttt{const}}$ ітерацій для підбору потрібного одноразового ключа.

Щодо методу №3 можна сказати, що він якнайкраще підходить нам із таких причин:

1) дана схема кардинально не змінює схему підпису та верифікації;

2) перевірка виконується за рахунок перевіряльника.

2.2 Запропонований метод усічення підпису для ДСТУ 4145-2002

Усічення підпису у даному алгоритмі буде відбуватися за допомогою упущення певної кількості біт із лівої частини підпису s без зміни схеми постановлення підпису. Тобто за основу візьмемо метод №3 із попереднього підпункту. Для виведення формул підсумую основні факти із 1.2.5:

1) Основна формула для перевірки правильності підпису – $R = sP + rQ$, де Q – відкритий ключ, P – базова точка, s,r – скаляри, T – вхідне повідомлення;

2) Межі для обрізання підпису: $8 \leqslant t \leqslant 32$;

3) Підпис D має наступну форму: $D = (r||s)$;

4) Останні перетворення перевіряння підпису мають наступну форму:

а) ...

б) $R = sP + rQ$, $R = (x_p, y_p)$

в) $y = hx_r$, $h = H(T)$

г) y перетворюється на ціле число r' ($r' = y_x$, де$y = (x_y, y_y)$)

д) $r \stackrel{?}{=} r'$

Саме у цьому стандарті додаткових покращень у вигляді нульових координат не будуть застосовані у зв'язку із особливостями формування підпису s. Позаяк упускаємо ліву частину підпису (старші біти), то підпис буде мати наступну форму:

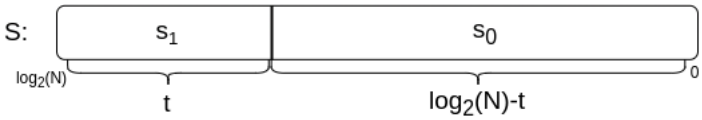

Рисунок 2.1 – Демонстрація поділу s на відому і шукану частини у ДСТУ 4145-2002.

Тобто:

$$s = s_0 + s_1 2^n \tag{2.1}$$

Алгоритм пошуку втраченої частини підпису ґрунтується на пошуку оригінальної частини серед великої кількості можливих варіантів. Цей процес можна описати наступним чином:

1) Відновлюємо точку R із уже відомого нам значення r.

2) Застосовуємо BSGS алгоритм [12], модифікований у нашому випадку.

3) Перевіряємо знайдену точку на коректність використовуючи дві рівності для перевірки. Припускаємо, що підпис є коректним тоді і тільки тоді, коли хоча б одна із рівностей буде істиною.

Повертаючись до виведення алгоритму, підставимо 2.1 у формулу 1),

тоді вийде наступне:

$$R = s_0 P + s_1(2^n P) + rQ \tag{2.2}$$

Наведімо спочатку алгоритм відновлення точки R напряму із даного числа r. Маємо те, що при постановленні підпису $r = h \cdot x_R$ обчислюється за допомогою операції у полі. Із цього виходить, що $x_R = r \cdot h^{-1}$ та, підставляючи у рівняння кривої(1.6), маємо 2 розв'язки рівняння:

$$-R = (x_R, x_R + y_R),\ R = (x_R, y_R).$$

Підсумуємо вивід у наступному алгоритмі:

Алгоритм 2.1. Алгоритм відновлення точки R для перевірки підпису згенерованого за схемою ДСТУ 4145-2002

Вхід:

– r – число у полі із якого будемо відновлювати точку ЕК R.

– a,b – параметри рівняння ЕК(1.6).

– h^{-1} – обернений елемент до результату гешування вхідного тексту.

Вихід: $R, -R$.

1) $x_R = r \cdot h^{-1}$ – операція у полі.

2) Підставляючи обчислене x_R у рівняння кривої $y^2+xy = x^3+ax^2+b$ знаходимо 2 точки $R, -R$.

Відповідно алгоритм узгодження для знаходження s_1 набуде такої форми:

Алгоритм 2.2. Алгоритм знаходження відповідного s_1 для відновлення підпису за схемою ДСТУ 4145-2002

1) Нехай I та J є додатними числами такі, що $IJ \geqslant 2^t$. Так як і раніше $J = 2^{t/2}$ та $I = 2^{t/2}$.

2) Для $j = 0$ до J, визначимо $U_j = s_0 P + jI(2^n P) + rQ$. Обчислюємо та накопичуємо координати x від точок U_j.

3) Для $i = 0$ до I, обчислюємо $V_i = R - i(2^n P)$ та перевіряємо чи координата x співпадає із одною у U_j.

На вихід отримуємо частину підпису для перевірки: $s_1 = i + Ij$.

Маючи алгоритм знаходження кандидата для перевірки та 2 відновлені точки $R, -R$, виведемо рівності для верифікації правильності підпису. Одразу зазначимо, що кадидат буде мати такий вигляд: $s = s_0 + (i + Ij)2^n$.

1) Підставляємо у формулу (2.2) результат виконання алгоритму 2.2 із припущенням, що точка R правильна:

$$\begin{aligned} s_0P + jI(2^nP) + rQ = R - i(2^nP) \Rightarrow \\ \Rightarrow R = (s_0 + s_1(2^n))P + rQ. \end{aligned} \tag{2.3}$$

2) Підставляємо у формулу (2.2) результат виконання 2.2 із припущенням, що точка $-R$ правильна:

$$\begin{aligned} s_0P + jI(2^nP) + rQ = -R - i(2^nP) \Rightarrow \\ \Rightarrow -R = (s_0 + s_1(2^n))P + rQ. \end{aligned} \tag{2.4}$$

У результаті отримуємо, що треба лише обчислити праву частину нерівності та порівнювати її із точками $R, -R$, що були відновлені. Відповідно до виведених двох формул можна отримати, що треба перевіряти один кадидат, який має форму: $s = s_0 + s_1 2^n$.

Зауваження. У модифікованому алгоритмі BSGS треба зберігати лише x координати для точок U_j, V_i. Тобто для обчислення використовуємо точки повністю і зберігаємо повноцінно їх, а для порівняння, лише x координати.

2.3 Запропонований метод усічення підпису за Ель-Гамалем

У даному підрозділі запропонуємо методи усічення підпису для класичної схеми Ель-Гамаля та його узагальнених варіантів. Запропоновані методи також ґрунтуються на підході №3 із попереднього

підрозділу. Для виведення формул нагадаю основні факти із схеми Ель-Гамаля 1.1.1:

Схема класичного Ель-Гамаля

1) r – відкритий ключ, g – елемент порядку q, s – скаляр або підпис, $H(m)$ – гешоване вхідне повідомлення.

2) Межі для обрізання підпису: $8 \leqslant t \leqslant 32$.

3) Підпис є парою чисел і має наступну форму: (r,s).

4) Останні перетворення встановлення підпису звичайною схемою має наступну форму:

а) ...

б) Перевірити $1 \leqslant r \leqslant p - 1$.

в) Перевірити нерівність:

$$y^r \cdot r^s \stackrel{?}{=} g^{H(m)} \bmod p. \tag{2.5}$$

Схема узагальненого Ель-Гамаля

Ця схема є певним узагальненням попередньої, тому більшість фактів однакові, окрім перевірки. Основні зміни наведені нижче.

1) Основне рівняння для вибору параметрів в узагальненій схемі є $Ak = B + Cx \pmod q$, де A, B, C – співвідношення від $H(m)$,r,s.

2) Останні перетворення встановлення підпису узагальненою схемою має форму:

а) ...

б) Перевірити $1 \leqslant r \leqslant p - 1$.

в) Перевірити нерівність:

$$r^A \stackrel{?}{=} g^B y^C \bmod p. \tag{2.6}$$

Запропоновані методи усічення підпису

Одразу зазначу, що на противагу алгоритму із ДСТУ 4145-2002 тут не буде відновлення точки, тому у даному випадку алгортим набуде такої форми.

1) Застосовуємо BSGS алгоритм [12], також модифікований, але по іншому у порівнянні із попереднім алгоритмом.

2) Перевіряємо знайдену відповідно до визначених співвідношеннях у алгоритмах.

Повертаючись до виведення алгоритмів для різних схем підпису Ель-Гамаля, то вони будуть мати наступну форму.

– **Для класичного Ель-Гамаля**

Розгляньмо реалізацію алгоритму верифікації для звичайної схеми Ель-Гамаля. Тут точно так само жодних покращень не буде, то ж припустимо, що s можна розділити на такі частини. Тобто:

$$s = s_0 + s_1 2^n. \tag{2.7}$$

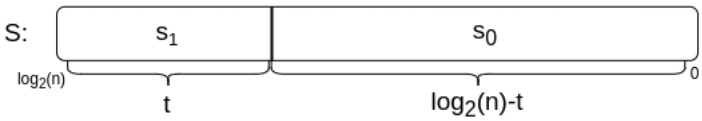

Рисунок 2.2 – Демонстрація поділу s на відому і шукану частини у Ель-Гамаля.

Перед наведенням алгоритмів, спочатку виведемо формули для обчислення. За умовою, знаємо, що $s_0, y, r, g, H(m)$ – дано. То ж підставимо (2.7) у (2.5):

$$y^r \cdot r^{s_0 + s_1 2^n} = g^{H(m)}. \tag{2.8}$$

Припустимо, що відповідь знайдемо у такому форматі $s_1 = i + jI$, тоді

$$y^r \cdot r^{s_0+i2^n+jI2^n} = g^{H(m)} \Rightarrow$$
$$y^r \cdot r^{s_0+i2^n} = g^{H(m)} \cdot r^{-jI2^n}. \quad (2.9)$$

Відповідно адаптований алгоритм BSGS для знаходження s_1 набуде наступної форми:

Алгоритм 2.3. Алгоритм знаходження відповідного s_1 для відновлення підпису за схемою класичного Ель-Гамаля

1) Нехай I та J є додатними числами, $J = 2^{t/2}$ та $I = 2^{t/2} - 1$.

2) Для $j = 0$ до J, визначимо $U_j = g^{H(m)} \cdot r^{-jI2^n}$. Обчислюємо та накопичуємо U_j.

3) Для $i = 0$ до I, обчислюємо $V_i = y^r \cdot r^{s_0+i2^n}$ та перевіряємо чи збігатися з одним U_j. Якщо є збіг, то ми маємо одного кандидата на перевірку: $s = s_0 + (i + Ij)2^n$. Далі перевіряємо його за формулою (2.8).

– **Для узагальненого Ель-Гамаля**

В аналогічний спосіб до попереднього будується алгоритм усічення цифрового підпису для узагальненої схеми Ель-Гамаля. Деякі із прикладів наведені у таблиці 2.1. Варто додати, що у таблиці параметри, позначені як «довільні», не повинні залежати від s; в іншому випадку наведений метод вимагатиме уточнення та додаткової адаптації. Наведімо вивід для одного випадку, де $A = s$, B, C — довільні.

Спочатку ділимо s на частини (2.7) та маємо подібне зображення 2.2. Відповідно до параметрів маємо формулу для перевірки підпису:

$$r^s = g^B y^C \pmod{p}. \quad (2.10)$$

Підставимо (2.7) у (2.10):

$$r^{s_0+s_1 2^n} = g^B y^C \pmod{p}. \quad (2.11)$$

Припустимо, що відповідь знайдемо у такому форматі $s_1 = i + jI \Rightarrow$

$$r^{s_0+i2^n+jI2^n} = g^B y^C \pmod{p} \Rightarrow$$
$$r^{s_0+i2^n} = g^B y^C r^{-jI2^n} \pmod{p}. \tag{2.12}$$

Відповідно алгоритм узгодження для знаходження s_1 матиме такий вигляд.

Алгоритм 2.4. Алгоритм знаходження відповідного s_1 для відновлення підпису за схемою узагальненого Ель-Гамаля, де $A = s$, B, C — довільні

1) Нехай I та J є додатними числами, $J = 2^{t/2}$ та $I = 2^{t/2} - 1$.

2) Для $j = 0$ до J, визначимо $U_j = g^B y^C r^{-jI2^n}$. Обчислюємо та накопичуємо U_j.

3) Для $i = 0$ до I, обчислюємо $V_i = r^{s_0+i2^n}$ та перевіряємо чи збігатися з одним U_j. Якщо є збіг, то ми маємо 1 кандидат на перевірку: $s = s_0 + (i + Ij)2^n$. Далі перевіряємо кандидата за формулою (2.11).

Зауважимо, що аналітична форма величин U_j, V_i визначається через параметри узагальненої схеми. Тоді узагальнена форма алгоритму для визначених параметрів буде мати такий вигляд:

Алгоритм 2.5. Алгоритм знаходження відповідного s_1 для відновлення підпису в узагальненій схемі

1) Нехай I та J є додатними числами, $J = 2^{t/2}$ та $I = 2^{t/2} - 1$.

2) Для $j = 0$ до J, обчислюємо та накопичуємо U_j.

3) Для $i = 0$ до I, обчислюємо V_i та перевіряємо, чи збігається з одним U_j. Якщо є збіг, то маємо кандидата на перевірку: $s = s_0 + (i + Ij)2^n$. Перевіряємо кандидата в залежності від вхідних параметрів.

Зауваження. Перший рядок у таблиці 2.1 є виведенням зі класичної схеми Ель-Гамаля 2.3. Також додам, що значення наведені у таблиці можна використовувати для виконання алгоритму перевірки укороченого підпису.

Таблиця 2.1 – Значення U_j, V_i для деяких варіантів узагальнених схем Ель-Гамаля із різними параметрами A,B,C

Параметри	**Перевірочне співвідношення**	U_j	V_i
$A = s$, $B = m$, $C = -r$	$r^s = g^m y^{-r}$	$g^m r^{-(jI2^n)}$	$y^r r^{(s_0+i2^n)}$
$A = s$, B, C — довільні	$r^s = g^B y^C$	$g^B y^C r^{-(jI2^n)}$	$r^{(s_0+i2^n)}$
$A = -s$, B, C — довільні	$r^{-s} = g^B y^C$	$g^B y^C r^{(jI2^n)}$	$r^{-(s_0+i2^n)}$
$B = s$, A, C — довільні	$r^A = g^s y^C$	$g^{(s_0+jI2^n)} y^C$	$r^A g^{-(i2^n)}$
$B = -s$, A, C — довільні	$r^A = g^{-s} y^C$	$g^{-(s_0+jI2^n)} y^C$	$r^A g^{(i2^n)}$
$C = s$, A, B — довільні	$r^A = g^B y^s$	$g^B y^{(s_0+jI2^n)}$	$r^A y^{i2^n}$
$C = -s$, A, B — довільні	$r^A = g^B y^{-s}$	$g^B y^{-(s_0+jI2^n)}$	$r^A y^{i2^n}$
$A = rs$, $B = m$, $C = \ const\ = c$	$r^{rs} = g^m y^c$	$g^m y^c r^{-(rjI2^n)}$	$r^{r(s_0+i2^n)}$
$A = ms$, B, C — довільні	$r^{ms} = g^B y^C$	$g^B y^C r^{-(mjI2^n)}$	$r^{m(s_0+i2^n)}$

2.4 Розроблення методів для перевірки

Було створено доповнення до комерційної бібліотеки [1], що була надана під час проходження практики у ТОВ «АВТОР». Відповідо до політики конфіденційності приклад програми не може бути наведений. Програмна реалізація була виконана на мові програмування C++ разом із інструментом для автоматичної збірки CMake, середовищем розробки Clion та вище зазначеною бібліотекою. Для наочності результатів наведемо приклад усічення підпису для кривих 163 та 431 біти у поліноміальному базисі.

1) **Крива 163 біти у поліноміальному базисі** Для виконання прикладу використовували:

– криву та базову точку визначену згідно стандарту [25];

– підписуване значення, що наведене у HEX кодуванні є рівним

```
0x09C9C44277910C9AAEE486883A2EB95.
B7180166DDF73532EEB76EDAEF52247FF;
```

– особистий ключ d, наведений у HEX кодуванні є рівним

`0x40000000000000002BEC12BE2262D39BCF14D` ;

– розмір підпису є рівний 42 байтам;

– розмір усічення є рівний 4 байтам.

Цифрові підписи для оригінального алгоритму та для усіченого алгоритму наведені у наступних картинках 2.3, 2.4.

```
original sign:
AE 79 CC 2A 3C A7 F6 12  BE C0 21 7D 02 EA 24 E4  |  .y.*<.....!}..$.
79 4E A9 69 00 85 93 78  49 94 19 D3 7F A0 0D B9  |  yN.i...xI.......
CB 44 DF 43 B3 D7 FA 3C  09 02                    |  .D.C...<..
```

Рисунок 2.3 – Цифровий підпис отриманиий оригінальним алгоритмом наведений у LSB форматі для кривої 163 біт.

```
truncated sign:
AE 79 CC 2A 3C A7 F6 12  BE C0 21 7D 02 EA 24 E4  |  .y.*<.....!}..$.
79 4E A9 69 00 85 93 78  49 94 19 D3 7F A0 0D B9  |  yN.i...xI.......
CB 44 DF 43 B3 D7                                 |  .D.C..
```

Рисунок 2.4 – Цифровий підпис отриманиий усіченим алгоритмом наведений у LSB форматі для кривої 163 біт.

Співпадіння було знайдено у згенерованих точок U_j, V_i із наступним кодуванням та індексами наведеними у наступних картинках 2.5, 2.6.

У результаті процедура отримує значення утраченої частини s_1 та відновлює підпис до оригінального 2.7, 2.8

На виході отримуємо, що усічений підпис є підтвердженим.

```
j = 521 = 0x209, U.j.x:
72 5B 7C 4F 81 DB CC 08  F7 70 48 91 B4 85 A7 E4  |  r[|O.....pH.....
EA 43 56 A6 06 00 00 00  00 00 00 00 00 00 00 00  |  .CV.............
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
```

Рисунок 2.5 – Отримане співпадіння у координаті x із обчисленої точки U_j на 521 ітерації для кривої 163 біт.

```
i = 15610 = 0x3CFA, V_i.x:
72 5B 7C 4F 81 DB CC 08  F7 70 48 91 B4 85 A7 E4  |  r[|O.....pH.....
EA 43 56 A6 06 00 00 00  00 00 00 00 00 00 00 00  |  .CV.............
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
```

Рисунок 2.6 – Отримане співпадіння у координаті x із обчисленої точки V_i на 15610 ітерації для кривої 163 біт.

```
s_1:
FA 3C 09 02 00 00 00 00  00 00 00 00 00 00 00 00  |  .<..............
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
```

Рисунок 2.7 – Отримане значення s_1, що обчислене із формули 3) для кривої 163 біт.

```
s:
85 93 78 49 94 19 D3 7F  A0 0D B9 CB 44 DF 43 B3  |  ..xI........D.C.
D7 FA 3C 09 02                                    |  ..<..
```

Рисунок 2.8 – Отримане вихідне значення підпису після конкатенації двох частин підпису для кривої 163 біт.

2) **Крива 431 біт у поліноміальному базисі**

Для виконання прикладу використовували:

– криву та базову точку визначену згідно стандарту [25];

– підписуване значення, що наведене у HEX кодуванні є рівним

0x09C9C44277910C9AAEE486883A2EB95B7180166DDF73532EEB76EDAEF52247FF;

– особистий ключ d, наведений у HEX кодуванні є рівним

0x40000000000000002BEC12BE2262D39BCF14D ;

– розмір підпису є рівний 108 байтам;

– розмір усічення є рівний 3 байтам.

Цифрові підписи для оригінального алгоритму та для усіченого алгоритму наведені у наступних картинках 2.9, 2.10.

```
original sign:
21 C0 75 35 03 65 FE 0A  48 E7 ED 0D A5 87 0B 33  |  !.u5.e..H......3
E0 9D 74 86 6E FB FF 00  E4 E5 F7 FB 77 0D DB 9D  |  ..t.n.......w...
47 FA E9 CC 84 AE C1 E4  E7 21 30 D6 A7 F2 54 EB  |  G........!0...T.
A1 CE 0F 1E D2 0C 4A 5E  49 FF 90 A2 6D 75 0A 22  |  ......J^I...mu."
64 2C 92 D6 8E 07 34 61  F6 C2 00 E8 17 D4 1B EB  |  d,....4a........
FD 72 A4 74 5E F0 2F 35  9D D5 E4 C4 88 D2 4B CC  |  .r.t^./5......K.
A5 3C B3 F9 78 9B 4A 1C  13 99 54 22              |  .<..x.J...T"
```

Рисунок 2.9 – Цифровий підпис отриманиий оригінальним алгоритмом наведений у LSB форматі для кривої 431 біт.

Співпадіння було знайдено у згенерованих точок U_j, V_i із наступним кодуванням та індексами наведеними у наступних картинках 2.11, 2.12.

У результаті процедура отримує значення утраченої частини s_1 та відновлює підпис до оригінального 2.13, 2.14.

На виході отримуємо, що усічений підпис є підтвердженим.

```
truncated sign:
21 C0 75 35 03 65 FE 0A  48 E7 ED 0D A5 87 0B 33  |  !.u5.e..H......3
E0 9D 74 86 6E FB FF 00  E4 E5 F7 FB 77 0D DB 9D  |  ..t.n.......w...
47 FA E9 CC 84 AE C1 E4  E7 21 30 D6 A7 F2 54 EB  |  G........!0...T.
A1 CE 0F 1E D2 0C 4A 5E  49 FF 90 A2 6D 75 0A 22  |  ......J^I...mu."
64 2C 92 D6 8E 07 34 61  F6 C2 00 E8 17 D4 1B EB  |  d,....4a........
FD 72 A4 74 5E F0 2F 35  9D D5 E4 C4 88 D2 4B CC  |  .r.t^./5......K.
A5 3C B3 F9 78 9B 4A 1C  13                       |  .<..x.J..
```

Рисунок 2.10 – Цифровий підпис отриманиий усіченим алгоритмом наведений у LSB форматі для кривої 431 біт.

```
j = 549 = 0x225, U_j.x:
6D 52 C2 65 DB E4 0E 83  DB C2 3A 93 4E BB 0E 7C  |  mR.e......:.N..|
77 FB 4F 84 5F FB E6 B0  36 FB 9D CC 1C 89 1F 51  |  w.O._...6......Q
9F 07 E4 55 02 2C 7E 9A  89 CB 60 E9 18 30 1F 01  |  ...U.,~...`..0..
F1 92 37 D7 0D 33 00 00  00 00 00 00 00 00 00 00  |  ..7..3..........
```

Рисунок 2.11 – Отримане співпадіння у координаті x із обчисленої точки U_j на 549 ітерації для кривої 431 біт.

```
i = 1177 = 0x499, V_i.x:
6D 52 C2 65 DB E4 0E 83  DB C2 3A 93 4E BB 0E 7C  |  mR.e......:.N..|
77 FB 4F 84 5F FB E6 B0  36 FB 9D CC 1C 89 1F 51  |  w.O._...6......Q
9F 07 E4 55 02 2C 7E 9A  89 CB 60 E9 18 30 1F 01  |  ...U.,~...`..0..
F1 92 37 D7 0D 33 00 00  00 00 00 00 00 00 00 00  |  ..7..3..........
```

Рисунок 2.12 – Отримане співпадіння у координаті x із обчисленої точки V_i на 1177 ітерації для кривої 431 біт.

Висновки до розділу 2

У розділі було запропоновано метод скорочення цифрового підпису для схем типу ДСТУ 4145-2002 та Ель-Гамаля. Були розглянуті ДСТУ 4145-2002, класична схема Ель-Гамаля та її узагальнений варіант. Запропоновані методи не змінюють процедуру постановки підпису;

```
s_1:
99 54 22 00 00 00 00 00  00 00 00 00 00 00 00 00  |  .T".............
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |  ................
```

Рисунок 2.13 – Отримане значення s_1, що обчислене із формули 3) для кривої 431 біт.

```
s:
4A 5E 49 FF 90 A2 6D 75  0A 22 64 2C 92 D6 8E 07  |  J^I...mu."d,....
34 61 F6 C2 00 E8 17 D4  1B EB FD 72 A4 74 5E F0  |  4a.........r.t^.
2F 35 9D D5 E4 C4 88 D2  4B CC A5 3C B3 F9 78 9B  |  /5......K..<..x.
4A 1C 13 99 54 22                                 |  J...T"
```

Рисунок 2.14 – Отримане вихідне значення підпису після конкатенації двох частин підпису для кривої 431 біт.

обчислений підпис скорочується шляхом відкидання певних бітів. Це дозволяє застосовувати запропоновані методи у реалізаціях криптографічних систем на малоресурсних пристроях, зокрема, використовувати вже існуючі реалізації без змін.

ВИСНОВКИ

У роботі було проведено порівняльний аналіз опублікованих методів усічення підписів типу Ель-Гамаля. Було продемонстровано, що методи, які ґрунтуються на використанні функції компресії та фіксуванні частини r або s, не застосовні, але метод із усіченням частини s задовольняє вимогам, які нас цікавлять. Використовуючи ідеї опублікованих методів для алгортимів ECDSA та EdDSA було запропоновано методи із усічення підписів для ДСТУ 4145-2002, а також для класичної та узагальнених схем цифрових підписів за Ель-Гамалем, які ґрунтуються на арифметиці великих чисел, а не на арифметиці еліптичних кривих. Проведено практичну перевірку запропонованих методів для еліптичних кривих над полем 163 та 431 біт. Показано високу ефективність при відсіканні до 32 бітів від оригінального підпису.

У подальших роботах треба досліджувати можливість покращення наведених алгоритмів, доцільність використання інших методів усічення, що не були реалізовані у даній роботі. Окрім цього треба досліджувати та розробляти методи зменшення підписів для еліптичних кривих.

ПЕРЕЛІК ПОСИЛАНЬ

[1] ТОВ «АВТОР». *Криптографічна бібліотека CryptoLib®V.3*. URL: https://avtor.ua/products/software/CryptoLibV3.

[2] Mehmet Adalier and Antara Teknik. "Efficient and secure elliptic curve cryptography implementation of curve p-256". In: *Workshop on elliptic curve cryptography standards*. Vol. 66. 2015, pp. 2014–2017.

[3] Liliya Akhmetzyanova et al. *On Methods of Shortening ElGamal-type Signatures*. Cryptology ePrint Archive, Paper 2021/148. 2021. URL: https://eprint.iacr.org/2021/148.

[4] Daniel J Bernstein et al. "High-speed high-security signatures". In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2011, pp. 124–142.

[5] Daniel J. Bernstein et al. *Twisted Edwards Curves*. Cryptology ePrint Archive, Paper 2008/013. https://eprint.iacr.org/2008/013. 2008. URL: https://eprint.iacr.org/2008/013.

[6] Antonin Dufka. "Schnorr Signatures with Application to Bitcoin". PhD thesis. Master's thesis, Masaryk University Faculty of Informatics, Czech Republic, 2020.

[7] Taher ElGamal. "A public key cryptosystem and a signature scheme based on discrete logarithms". In: *IEEE transactions on information theory* 31.4 (1985), pp. 469–472.

[8] *FIPS PUB 186-4: Digital signature standard (DSS)*. 2013. URL: https://csrc.nist.gov/publications/detail/fips/186/4/final.

[9] Don Johnson, Alfred Menezes, and Scott Vanstone. "The elliptic curve digital signature algorithm (ECDSA)". In: *International journal of information security* 1.1 (2001), pp. 36–63.

[10] Neal Koblitz and Alfred J. Menezes. *A Riddle Wrapped in an Enigma*. Cryptology ePrint Archive, Paper 2015/1018. https://eprint.iacr.org/2015/1018. 2015. URL: https://eprint.iacr.org/2015/1018.

[11] Illia Kripaka and Serhii Yakovliev. "Truncated DSTU 4145-2002 Digital Signatures". In: *Matherials of First International Scientific and Practical Conference «Cyberwarfare: Intelligence, Defence and Offensive Security» (April 20 – 21, Kyiv, Ukraine)*. Kyiv: MITIT, 2023, pp. 10–11.

[12] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996, p. 816. ISBN: 0-8493-8523-7.

[13] Gregory Neven, Nigel P Smart, and Bogdan Warinschi. "Hash function requirements for Schnorr signatures". In: *Journal of Mathematical Cryptology* 3.1 (2009), pp. 69–87.

[14] Cesar Pereida Garcia and Sampo Sovio. "Size, Speed, and Security: An Ed25519 Case Study". In: *Nordic Conference on Secure IT Systems*. Springer. 2021, pp. 16–30.

[15] Thomas Pornin. *Truncated EdDSA/ECDSA Signatures*. Cryptology ePrint Archive, Paper 2022/938. https://eprint.iacr.org/2022/938. 2022. URL: https://eprint.iacr.org/2022/938.

[16] Bruce Schneier. *Applied Cryptography: Protocols, Algorthms, and Source Code in C. 20th Anniversary Edition*. John Wiley & Sons, 2015, p. 784. ISBN: 978-1-119-09672-6.

[17] Claus P. Schnorr. *Method for identifying subscribers and for generating and verifying electronic signarutes in a data exchange*. 1991. URL: https://patentimages.storage.googleapis.com/4c/8b/54/6eda154a2e0627/US4995082.pdf.

[18] Peter W Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134.

[19] FEDERAL INFORMATION PROCESSING STANDARDS. *Digital Signature Standard (DSS), FIPS PUB 186*. 1994. URL: https://csrc.nist.gov/publications/detail/fips/186/archive/1996-12-30.

[20] FEDERAL INFORMATION PROCESSING STANDARDS. *Digital Signature Standard (DSS), FIPS PUB 186-5*. 2019. URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5-draft.pdf.

[21] Loup Vaillant. *Cofactor Explained: Clearing Elliptic Curves' dirty little secret*. 2020. URL: https://loup-vaillant.fr/tutorials/cofactor.

[22] Є. А. Архипська. "Дослідження прихованих каналів передачі даних в ДСТУ 4145-2002". In: *XVII Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики»*. 2019, pp. 192–194. URL: https://ela.kpi.ua/handle/123456789/29797.

[23] Громадський простір. *Відкритий тендер із закупівлі: «Робоча станція для оформлення та видачі паспортних документів, що підтверджують громадянство України, посвідчують особу чи її спеціальний статус, з комплектом обладнання для зняття біометричних даних (параметрів) особи»*. 2022. URL: https://www.prostir.ua/?grants=vidkrytyj-tender-iz-zakupivli-robocha-stantsiya-dlya-oformlennya-ta-vydachi-pasportnyh-dokumentiv-scho-pidtverdzhuyut-hromadyanstvo-ukrajiny-posvidchuyut-osobu-chy-jiji-spetsialnyj-status-z-komple.

[24] Ілля Кріпака. "Усічення цифрового підпису для схем типу Ель-Гамаля". українська. In: *XXI Всеукраїнська науково-практична конференція «Теоретичні і прикладні проблеми фізики, математики та інформатики» (Травень 11 – 12, Київ, Україна)*. Київ: НН ФТІ, 2023, pp. 206–208.

[25] *Національний стандарт України. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка ДСТУ 4145-2002.* 2003.

[26] *Про встановлення вимог до технічних засобів, процесів їх створення, використання та функціонування у складі інформаційно-телекомунікаційних систем під час надання кваліфікованих електронних довірчих послуг.* МІНІСТЕРСТВО ЦИФРОВОЇ ТРАНСФОРМАЦІЇ УКРАЇНИ. Україна, 2020. URL: https://zakon.rada.gov.ua/laws/show/z1039-20.

[27] Ю. Яремчук. *Метод цифрового підписування на основі математичного апарату еліптичних кривих з прискореною процедурою перевірки підпису.* 1ий. 2007, pp. 118–127. URL: https://ela.kpi.ua/handle/123456789/10677.