

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

«На правах рукопису»

УДК 004.056.55

«До захисту допущено»

Завідувач кафедри

_____ Сергій ЯКОВЛЄВ

«__» _____ 2026 р.

Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою
«Математичні методи криптографічного захисту інформації»
зі спеціальності: 113 Прикладна математика
на тему: «Застосування схем цифрового підпису Штерна і
Верона в малоресурсних пристроях»

Виконав:

студент IV курсу, групи ФІ-23

Пацьора Поліна Олегівна _____

Керівник:

ст. викл. каф. ММЗІ, к.ф.-м.н.

Фесенко Андрій В'ячеславович _____

Рецензент:

доц. каф. ММАД, доктор філософії

Яйлимова Ганна Олексіївна _____

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

Студент _____

**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

Завідувач кафедри

_____ Сергій ЯКОВЛЄВ

«__» _____ 2026 р.

**ЗАВДАННЯ
на дипломну роботу**

Студент: Пацьора Поліна Олегівна

1. Тема роботи: *«Застосування схем цифрового підпису Штерна і Верона в малоресурсних пристроях»*, науковий керівник дисертації: ст. викл. каф. ММЗІ, к.ф.-м.н. Фесенко Андрій В'ячеславович,

затверджені наказом по університету №__ від «__» _____ 2026 р.

2. Термін подання студентом роботи: «__» _____ 2026 р.

3. Об'єкт дослідження: постквантові криптографічні механізми забезпечення автентичності та цілісності даних в умовах обмежених обчислювальних ресурсів.

4. Предмет дослідження: схеми цифрового підпису Штерна і Верона, їхні параметри, розміри відкритих і секретних ключів, розміри підписів та особливості застосування в малоресурсних пристроях.

5. Перелік завдань:

1) провести огляд опублікованих джерел, що досліджують застосування схем цифрового підпису Штерна в малоресурсних пристроях;

2) дослідити принципи побудови схеми підпису Штерна та наявні модифікації ;

- 3) побудувати модифікацію схеми цифрового підпису Верона;
- 4) виконати порівняльний аналіз схем Штерна і Верона за розмірами відкритих ключів, секретних ключів та підписів ;
- 5) провести аналітичну оцінку ресурсної складності застосування схем Штерна і Верона в малоресурсних пристроях.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу:
Презентація доповіді
7. Орієнтовний перелік публікацій: планується доповідь на всеукраїнській конференції
8. Дата видачі завдання: 10 вересня 2025 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	01-15 вересня 2025 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	Вересень-жовтень 2025 р.	Виконано
3	Дослідження схеми ідентифікації Штерна та принципів побудови Штерн-подібних схем підпису	Листопад-грудень 2025 р.	Виконано
4	Аналіз модифікацій схем підпису Штерна та Верона	Січень-лютий 2026 р.	Виконано
5	Побудова модифікації схеми цифрового підпису Верона	Березень 2026 р.	Виконано
6	Аналіз параметрів схем Штерна і Верона: розмірів відкритих і секретних ключів, підписів та кількості раундів	Квітень 2026 р.	Виконано
7	Оцінка ресурсної складності застосування схем у малоресурсних пристроях	Травень 2026 р.	Виконано
8	Оформлення дипломної роботи	Початок червня 2026 р.	Виконано

Студент

_____ Поліна ПАЦЬОРА

Керівник

_____ Андрій ФЕСЕНКО

РЕФЕРАТ

Кваліфікаційна робота містить: 48 стор., 1 таблицю, 19 джерел.

Метою роботи є визначення можливості та умов практичної реалізації схем цифрового підпису Штерна і Верона на малоресурсних пристроях.

Об'єктом дослідження є процеси перетворення інформації в умовах обмежених обчислювальних ресурсів.

Предметом дослідження є властивості схеми цифрового підпису Штерна і Верона та особливості застосування в малоресурсних пристроях.

У роботі розглянуто особливості побудови схеми цифрового підпису Штерна та Верона й їх модифікації. Побудовано модифікацію схеми цифрового підпису Верона. Виконано порівняльний аналіз схем за розмірами ключів і підпису та аналітичну оцінку їх ресурсної складності.

ПОСТКВАНТОВА КРИПТОГРАФІЯ, ЦИФРОВИЙ ПІДПИС,
СХЕМА ШТЕРНА, СХЕМА ВЕРОНА, ДЕКОДУВАННЯ СИНДРОМУ,
МАЛОРЕСУРСНІ ПРИСТРОЇ.

ABSTRACT

The qualification work contains: 48 pages, 1 table, 19 sources.

The aim of the work is to determine the feasibility and conditions of practical implementation of the Stern and Véron digital signature schemes on resource-constrained devices.

The research object is information transformation processes under conditions of limited computational resources.

The research subject is the properties of the Stern and Véron digital signature schemes and the specific features of their application in resource-constrained devices.

The paper reviews the construction principles of the Stern and Véron digital signature schemes and their modifications. One of the modification was applied to the Véron digital signature scheme. A comparative analysis of the schemes by key and signature sizes was performed, as well as an analytical evaluation of their resource complexity.

POST-QUANTUM CRYPTOGRAPHY, DIGITAL SIGNATURE,
STERN SCHEME, VÉRON SCHEME, SYNDROME DECODING,
RESOURCE-CONSTRAINED DEVICES.

ЗМІСТ

Перелік умовних позначень, скорочень і термінів	8
Вступ.....	9
1 Перетворення схеми ідентифікації Штерна в схему цифрового підпису	11
1.1 Структура та властивості схеми ідентифікації з нульовим розголошенням	11
1.2 Опис схеми ідентифікації Штерна	14
1.3 Схеми підпису, побудовані на основі схеми ідентифікації	17
Висновки до розділу 1	19
2 Дослідження модифікацій схем підпису Штерна і Верона	20
2.1 Зменшення розміру підпису в схемі Штерна засобами псевдовипадкової генерації.....	20
2.2 Атака на схему підпису Штерна з детермінованими зобов'язаннями	23
2.3 Усунення вразливості оптимізованої схеми підпису за допомогою рандомізації зерен	25
2.4 Альтернативні схеми на основі квазіциклічних кодів	27
2.5 Схема ідентифікації Верона	29
2.6 Модифікація протоколу Верона до рангової метрики на основі подвійних циркулянтних кодів	31
Висновки до розділу 2	35
3 Використання схем Штерна і Верона для пристроїв з обмеженими ресурсами	36
3.1 Побудова модифікації схеми цифрового підпису Верона	36
3.2 Вимоги до постквантового підпису для IoT	39
3.3 Вплив параметрів схем Штерна та Верона на ресурси пристроїв .	40
Висновки до розділу 3	44
Висновки	45
Перелік посилань	46

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

$\xleftarrow{\$}$ — обирається випадково (рівномірно)

$w_H(x)$ — вага Хеммінга

$\mathbb{F}_2$ — поле лишків за модулем 2

trans — запис виконання протоколу

QC — квазіциклічний (*quasi-cyclic*)

$\mathcal{H}$ — геш-функція

ВСТУП

Актуальність дослідження. На сьогодні малоресурсні пристрої використовуються майже в будь-якій сфері, наприклад: медицина, логістика, системи автоматизації, фінансові сервіси, тощо. Відповідно виникає необхідність у контролі цілісності даних та забезпеченні автентифікації джерел цих даних, і для досягнення цього використовується цифровий підпис.

Водночас традиційні схеми підпису, стійкість яких використовує задачі факторизації та дискретного логарифмування не є стійкими до квантових алгоритмів. У зв'язку з цим особливої актуальності набуває дослідження постквантових схем цифрового підпису, стійкість яких ґрунтується на задачах, що вважаються складними як для класичних, так і для квантових обчислювальних моделей і одним із перспективних напрямів постквантової криптографії є кодова криптографія, яка ґрунтується на складності задач теорії кодування. До таких підходів належать схеми ідентифікації та цифрового підпису, побудовані на основі протоколів Штерна і Верона. Ці схеми є важливими об'єктами дослідження, оскільки поєднують постквантову стійкість із відносно простою алгебраїчною структурою, що робить їх потенційно придатними для реалізації у вбудованих системах.

Метою дослідження є визначення можливості та умов практичної реалізації схем цифрового підпису Штерна і Верона на малоресурсних пристроях..

Для досягнення мети необхідно вирішити такі завдання:

- 1) провести огляд опублікованих джерел, що досліджують застосування схем цифрового підпису Штерна в малоресурсних пристроях;
- 2) дослідити принципи побудови схеми підпису Штерна та наявні модифікації ;

- 3) побудувати модифікацію схеми цифрового підпису Верона
- 4) виконати порівняльний аналіз схем Штерна і Верона за розмірами відкритих ключів, секретних ключів та підписів ;
- 5) провести аналітичну оцінку ресурсної складності застосування схем Штерна і Верона в малоресурсних пристроях.

Об'єктом дослідження є процеси перетворення інформації в умовах обмежених обчислювальних ресурсів.

Предметом дослідження є властивості схеми цифрового підпису Штерна і Верона та особливості застосування в малоресурсних пристроях.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: методи лінійної та абстрактної алгебри, методи теорії кодування, методи комбінаторного аналізу, методи теорії складності алгоритмів, аналітичне оцінювання параметрів криптографічних схем, порівняльний аналіз, а також методи математичного моделювання для оцінювання розмірів ключів, підписів і обсягу пам'яті.

Наукова новизна отриманих результатів полягає в порівняльному аналізі параметрів схем цифрового підпису Штерна і Верона з погляду їх застосування в малоресурсних пристроях. У роботі побудовано оптимізацію схеми цифрового підпису Верона; виконано аналітичну оцінку розмірів відкритих і секретних ключів, підписів, кількості раундів

Практичне значення результатів полягає в тому, що отримані оцінки можуть бути використані під час вибору параметрів схем цифрового підпису для пристроїв з обмеженими ресурсами. Проведений аналіз дозволяє визначити основні обмежувальні фактори під час реалізації схем Штерна і Верона, зокрема розмір відкритого ключа, розмір підпису, кількість раундів і потребу в пам'яті.

1 ПЕРЕТВОРЕННЯ СХЕМИ ІДЕНТИФІКАЦІЇ ШТЕРНА В СХЕМУ ЦИФРОВОГО ПІДПISУ

У цьому розділі розглянемо схеми ідентифікації та схеми підпису, що походять від протоколу ідентифікації, запропонованого Ж. Штерном [18]. Їхня безпека ґрунтується на складності задачі декодування синдрому та використання перетворення Фіата–Шаміра [10]. Головною проблемою оригінальної схеми підпису Штерна, як і більшості кодових схем є надмірно великий розмір підпису, що може досягати кількох сотень кілобайтів.

1.1 Структура та властивості схеми ідентифікації з нульовим розголошенням

Схеми ідентифікації є надзвичайно корисними та базовими інструментами в багатьох застосуваннях, таких як електронні фінансові перекази та онлайн-системи, що запобігають доступу до даних недійсних користувачів. Такі схеми є типовими прикладами інтерактивних доведень із нульовим розголошенням - двосторонніх протоколів, у яких одна сторона, що називається доказувачем, може переконати іншу сторону, перевіряльника, у тому, що вона знає певну секретну інформацію, причому перевіряльник не дізнається нічого про це секретне значення, окрім того, що доказувач розкриває сам.

Означення 1.1. 3-раундова схема ідентифікації — це інтерактивний протокол, що складається з 3 алгоритмів, які мають бути ефективними відносно деякого параметра безпеки $\lambda \in \mathbb{N}$:

- алгоритм створення ключів $\text{Keygen}(1^\lambda) \rightarrow (pk, sk)$
- алгоритми доказувача $P = (P_1, P_2)$, для яких виконується $P_1(sk) \rightarrow (x, St)$ та $P_2(sk, x, c, St) \rightarrow z$, де x — перше повідомлення, St

- деякий внутрішній стан, c — виклик (*challenge*) від перевіряльника, та $z \in R$ — відповідь доказувача (тобто друге повідомлення),
- алгоритм перевіряльника $V(pk, x, c, z)$, який виводить 1, якщо доказ приймається, і 0 — інакше.

Далі розглянемо основні етапи схеми ідентифікації

Ініціалізація. Нехай $(pk, sk) \leftarrow \text{Keygen}(1^\lambda)$. Доказувач отримує (pk, sk) , а перевіряльник отримує pk .

Обмін.

- 1) Доказувач обчислює $(x, St) \leftarrow P_1(sk)$ і надсилає x перевіряльнику.
- 2) Перевіряльник вибирає випадкове c з множини викликів C і надсилає його доказувачу.
- 3) Доказувач обчислює $z \leftarrow P_2(sk, x, c, St)$ і надсилає z перевіряльнику.

Перевірка. Перевіряльник приймає доказ тоді і тільки тоді, коли $V(pk, x, c, z) = 1$.

Запис виконання протоколу (далі - транскрипт) складається з повідомлень якими обмінюються доказувач та перевіряльник за запуску схеми ідентифікації,.

Означення 1.2. Розподіли транскриптів, чесна поведінка [8]

Для схеми ідентифікації $\text{trans}_{IS}(pk, sk)$, є розподіл транскриптів, коли і доказувач, і перевіряльник діють чесно. Цей розподіл визначається так:

$$\text{trans}_{IS}(pk, sk) : (x, St) \leftarrow P_1(sk), \quad c \leftarrow C, \quad z \leftarrow P_2(sk, x, c, St), \quad \text{return } (x, c, z).$$

Також розглянемо транскрипти у випадку, коли зловмисник $\mathcal{A}$, який знає лише pk , намагається видавати себе за доказувача.

Твердження 1.1. [8]

Для схеми ідентифікації IS та зловмисника $\mathcal{A}(pk)$, $\text{trans}_{IS}^{\mathcal{A}}(pk)$, є розподіл

транскриптів, що виникає тоді, коли зловмисник $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, який знає лише відкритий ключ, взаємодіє з перевіряльником.

$$\text{trans}_{IS}^{\mathcal{A}}(pk) : \quad (\tilde{x}, \tilde{St}) \leftarrow \mathcal{A}_1(pk), \quad c \leftarrow C, \quad \tilde{z} \leftarrow \mathcal{A}_2(pk, x, c, \tilde{St}), \quad \text{return } (\tilde{x}, c, \tilde{z}).$$

Ідентифікація має задовільняти властивості повноти, коректності та нульового розголошення (zero-knowledge).

Означення 1.3. Повнота [8]

Схему ідентифікації IS називають досконало повною тоді й лише тоді:

$$\Pr \left[V(pk, x, c, z) = 1 \mid \begin{array}{l} (pk, sk) \leftarrow \text{Keygen}(1^\lambda), \\ (x, c, z) \leftarrow \text{trans}_{IS}(pk, sk) \end{array} \right] = 1.$$

Означення 1.4. Перевага зловмисника в коректності [8]

Нехай IS — схема ідентифікації. Для будь-якого зловмисника $\mathcal{A}$ визначимо ймовірність успішного обману перевіряльника :

$$\text{Adv}_{IS}(\mathcal{A}) = \Pr \left[V(pk, x, c, z) = 1 \mid \begin{array}{l} (pk, sk) \leftarrow \text{Keygen}(1^\lambda), \\ (x, c, z) \leftarrow \text{trans}_{IS}^{\mathcal{A}}(pk) \end{array} \right].$$

Властивість нульового розголошення означає, що під час виконання схеми ідентифікації перевіряльник не отримує жодної додаткової інформації про секретний ключ. Тобто, навіть маючи записи виконання протоколу, неможливо відновити секретні дані або отримати відомості, які допомогли б у їх знаходженні. В роботі [8] ця властивість визначається через існування ефективного симулятора $\text{Sim}(pk)$, здатного виводити записи виконання протоколу, які є обчислювально нерозрізнюваними від записів, отриманих під час реального виконання протоколу. Для заданого симулятора величина $\text{Adv}_{IS}^{HVZK}(\mathcal{A})$ відповідає ймовірності того, що зловмисник $\mathcal{A}$ зможе розрізнити реальний та

симульований записи виконання протоколу.

Означення 1.5. [8]

Нехай IS — схема ідентифікації, для якої зафіксовано симулятор Sim , тоді для будь-якого зломисника $\mathcal{A}$ визначається ймовірність того, що перевіряльник прийме коректне виконання протоколу між чесними сторонами:

$$\text{Adv}_{IS}^{HVZK}(\mathcal{A}) = \left| \Pr \left[b = 1 \mid \begin{array}{l} (pk, sk) \leftarrow \text{Keygen}(1^\lambda), \\ \tau = (x, c, z) \leftarrow \text{trans}_{IS}(pk, sk), \\ b \leftarrow \mathcal{A}(\tau, pk) \end{array} \right] - \Pr \left[b = 1 \mid \begin{array}{l} (pk, sk) \leftarrow \text{Keygen}(1^\lambda), \\ \tau = (x, c, z) \leftarrow \text{Sim}(pk), \\ b \leftarrow \mathcal{A}(\tau, pk) \end{array} \right] \right|$$

1.2 Опис схеми ідентифікації Штерна

Перша схема ідентифікації з нульовим розголошенням побудована на теорії кодування була представлена Штерном [18] на конференції Crypto'93; її безпека ґрунтується на складності задачі декодування синдрому (SD).

Задача SD була доведена як NP-повна в [2]

Задача 1.1. Декодування синдрому, SD-problem [3]

Нехай задано додатні цілі числа n, k, w , випадкову матрицю перевірки парності $H \xleftarrow{\$} \mathbb{F}_2^{(n-k) \times n}$, і синдром $s \in \mathbb{F}_2^{n-k}$. Тоді задача декодування синдрому $SD(n, k, w)$ полягає у знаходженні вектора $e \in \mathbb{F}_2^n$, такого, що $He^\top = s^\top$ і $w_H(e) = w$.

Твердження 1.2. [8]

Для відповідного вибору параметрів $R = \frac{k}{n} = \Theta(1)$ та $\omega = \frac{w}{n} = \Theta(1)$, задача $SD(n, k, w)$, визначена відносно ваги Хеммінга $wt(\cdot)$, є складною у середньому, коли вхідні дані (H, s) вибираються згідно з розподілом $\mathcal{D}_{n, k, w}$,

заданим:

$$\mathcal{D}_{n,k,w} : \quad H \xleftarrow{\$} \mathbb{F}_q^{(n-k) \times n}, \quad e \xleftarrow{\$} S_w, \quad \text{return } (H, s = He).$$

де S_w — множина всіх векторів ваги w .

Далі введемо означення схеми зобов'язання. Схема зобов'язання потрібна в протоколі ідентифікації для того, щоб доказувач не міг змінювати свої повідомлення після отримання виклику, водночас не розкриваючи секретну інформацію, що забезпечує коректність і нульове розголошення протоколу.

Означення 1.6. [8] Схема зобов'язання складається з трьох алгоритмів:

- $\text{Commit}(s) \rightarrow (c, St)$, де c - зобов'язанням, а St — деякий внутрішній стан, що використовується для відкриття.
- $\text{Open}(s, St) \rightarrow o$, що виводить відкриття o для рядка s .
- $\text{Ver}(c, s, o) \rightarrow \{0,1\}$, що перевіряє правильність відкриття o для рядка s та зобов'язання c .

Для зобов'язання c відкриття (s,o) є коректним тоді й лише тоді, коли $\text{Ver}(c,s,o) = 1$.

Означення 1.7. 1-раундова схема ідентифікації Штерна, загальне зобов'язання, [8]

Ініціалізація. Нехай $(H, e, s = He) \xleftarrow{\$} \mathcal{D}_{n,k,w}$, $sk = e$, $pk = (H, s)$. Доказувач отримує (pk, sk) , а перевіряльник отримує pk .

Обмін.

– Доказувач вибирає випадкову перестановку $\pi \in \text{Perm}_n$ та випадковий вектор-стовпчик $y \xleftarrow{\$} \mathbb{F}_q^n$, після чого обчислює $t = Hy$. Доказувач обчислює $z_1 = (\pi, t)$, $z_2 = \pi(y)$, $z_3 = \pi(y + e)$. Для $i \in \{1,2,3\}$ він формує зобов'язання $(x_i, St_i) \leftarrow \text{Commit}(z_i)$ та надсилає перевіряльнику x_1, x_2, x_3 .

– Перевіряльник надсилає доказувачу випадковий виклик $c \xleftarrow{\$} \{1,2,3\}$.

– Доказувач розкриває інформацію так, щоб перевіряльник міг відновити z_i для всіх $i \in \{1,2,3\} \setminus \{c\}$. Більш точно:

1) Якщо $c = 1$, доказувач обчислює $o_2 = \text{Open}(z_2, St_2)$, $o_3 = \text{Open}(z_3, St_3)$, і надсилає перевіряльнику z_2, z_3, o_2, o_3 .

2) Якщо $c = 2$, доказувач обчислює $o_1 = \text{Open}(z_1, St_1)$, $o_3 = \text{Open}(z_3, St_3)$, і надсилає перевіряльнику z_1, z_3, o_1, o_3 .

3) Якщо $c = 3$, доказувач обчислює $o_1 = \text{Open}(z_1, St_1)$, $o_2 = \text{Open}(z_2, St_2)$, і надсилає перевіряльнику z_1, z_2, o_1, o_2 .

Перевірка.

1) Якщо $c = 1$, перевіряльник перевіряє, що $|z_2 + z_3| = w$, а також перевіряє коректність зобов'язань z_2 та z_3 , тобто що $Ver(x_2, z_2, o_2) = Ver(x_3, z_3, o_3) = 1$.

2) Якщо $c = 2$, перевіряльник отримує $z_1 = (\pi, t)$ та $z_3 = \pi(y + e)$. Він перевіряє, що $H(\pi^{-1}(z_3)) = H(y + e) = t + s$. Також він перевіряє коректність зобов'язань z_1 та z_3 .

3) Якщо $c = 3$, перевіряльник отримує $z_1 = (\pi, t)$ та $z_2 = \pi(y)$. Він перевіряє, що $H(\pi^{-1}(z_2)) = H(y) = t$. Також він перевіряє коректність зобов'язань z_1 та z_2 .

Для підвищення рівня захищеності схема Штерна будується багаторазовим паралельним виконанням однораундової схеми, так як в межах одного раунду ймовірність зломисника пройти перевірку без знання секретного вектора - становить $\frac{2}{3}$. Кожне додаткове повторення зменшує ймовірність успішного обману, тому загальна стійкість схеми визначається кількістю раундів R . Якщо ймовірність обману в одному раунді дорівнює $\frac{2}{3}$, то після R незалежних повторень ймовірність становитиме $(\frac{2}{3})^R$. Для забезпечення рівня безпеки λ кількість раундів обирають так, щоб ймовірність успішної підробки була не більшою за $2^{-\lambda}$, тобто $(\frac{2}{3})^R \leq 2^{-\lambda}$.

У випадку використання квантового комп'ютера зломисником

кількість раундів подвоюється, оскільки квантові алгоритми можуть підвищувати ефективність атак. Таким чином, рівень безпеки схеми Штерна безпосередньо визначається ймовірністю успішного обману в одному раунді та кількістю паралельних повторень протоколу.

1.3 Схеми підпису, побудовані на основі схеми ідентифікації

Означення 1.8. [5] **Схема підпису** $S = (G, S, V)$ це трійка ефективних алгоритмів G, S, V , де G - алгоритм створення ключів, S - алгоритм підписування, V - алгоритм перевірки.

- G - ймовірнісний алгоритм без вхідних даних. Він створює пару (pk, sk)
- S - ймовірнісний алгоритм, що сформує підпис $\sigma \xleftarrow{\$} S(sk, m)$
- V - детермінований алгоритм, що перевіряє правильність підпису і повертає `accept` або `reject`.

Схема підпису є коректною, тоді й тільки тоді, якщо для $(pk, sk) \leftarrow G$ і для кожного повідомлення m , маємо

$$\Pr[V(pk, m, S(sk, m)) = \text{accept}] = 1.$$

Використовуючи перетворення Фіата–Шаміра, можна перетворити схеми ідентифікації на схеми цифрового підпису. Основна ідея полягає в тому, що підписувач діє як доказувач, запускаючи схему ідентифікації самотійно. Тобто щоб підписати повідомлення m , підписувач спочатку обчислює початкове повідомлення x , а потім генерує виклик c застосовуючи геш-функцію функцію $\mathcal{H}$ до x та m . Далі формується правильна відповідь s . Підписом повідомлення m є пара (c, z) , яку можна перевірити, (1) повторно обчисливши $x = \mathcal{V}(pk, c, z)$ та (2) перевіривши, що $\mathcal{H}(x, m) \stackrel{?}{=} c$.

Означення 1.9. Перетворення Фіата–Шаміра [13]

Нехай $(\text{Keygen}_{IS}, \mathcal{P}_1, \mathcal{P}_2, \mathcal{V})$ є схемою ідентифікації. Побудуємо схему підпису таким чином:

– *Keygen_S*: на вхід приймається 1^λ . Запускається $\text{Keygen}_{IS}(1^\lambda)$ та генеруються ключі (pk, sk) . Відкритий ключ pk задає множину викликів Ω_{pk} . Як частина процедури генерування ключів задається функція $\mathcal{H} : \{0,1\}^* \rightarrow \Omega_{pk}$, проте надалі вважатимемо її заданою неявно

– **Sign**: на вхід приймається приватний ключ sk та повідомлення $m \in \{0,1\}^*$.

1) Обчислити $(x, st) \leftarrow \mathcal{P}_1(sk)$.

2) Обчислити $c := \mathcal{H}(x, m)$.

3) Обчислити $z := \mathcal{P}_2(sk, st, c)$.

Повертається підпис (c, z) .

– **Ver**: на вхід приймається відкритий ключ pk , повідомлення m та підпис (c, z) . Обчислюється $x := \mathcal{V}(pk, c, z)$ та повертається 1 тоді й тільки тоді, якщо

$$\mathcal{H}(x, m) \stackrel{?}{=} c.$$

Зауваження. [13]

Підпис (c, z) є “прив’язаним” до певного повідомлення m , оскільки r є функцією як від x , так і від m ; відповідно, зміна m приводить до повністю іншого значення c . Якщо $\mathcal{H}$ змодельовано як випадковий оракул, що відображає вхідні дані рівномірно в Ω_{pk} , тоді виклик r є рівномірним. Інтуїтивно це означає, що зловмиснику, який не знає секретного ключа sk буде так само важко знайти вірний підпис (c, z) для повідомлення m , як і видати себе за доказувача під час чесного виконання протоколу ідентифікації.

Теорема 1.1. [13]

Нехай Π є схемою ідентифікації, а Π' — схемою підпису, отриманою шляхом застосування до неї перетворення Фіата–Шаміра. Якщо Π є захищеною, а $\mathcal{H}$ змодельовано як випадковий оракул, то Π' також є захищеною.

Висновки до розділу 1

У цьому розділі розглянуто кодові схеми ідентифікації, побудовані на основі протоколу Штерна, а також спосіб перетворення на схему цифрового підпису за допомогою перетворення Фіата-Шаміра, наведено їхні основні криптографічні властивості, такі як коректність, повнота, стійкість до підробки. Також було показано, що безпека Штерн-подібних схем ґрунтується на складності задачі декодування синдрому. Отримані теоретичні відомості можуть бути використані як теоретична основа для подальшого вивчення оптимізованих кодових Штерн-подібних схем підпису.

2 ДОСЛІДЖЕННЯ МОДИФІКАЦІЙ СХЕМ ПІДПISУ ШТЕРНА І ВЕРОНА

У цьому розділі досліджуються модифікації протоколів ідентифікації та цифрового підпису, що будуються на схемі Штерна і схемі Верона. Розглядаються методи оптимізації комунікаційної складності, побудовані на використанні псевдовипадкових генераторів та детермінованих комітментів, а також аналізується їх вплив на криптографічну стійкість відповідних схем. Також окремо розглянуто схеми на основі квазіциклічних кодів та схему Верона як дуальну версію протоколу Штерна і її узагальнення на рангову метрику.

2.1 Зменшення розміру підпису в схемі Штерна засобами псевдовипадкової генерації

Протоколи ідентифікації на основі схеми Штерна характеризуються значним обсягом переданих даних, зумовленим необхідністю передачі повних випадкових векторів на кожному раунді.

З метою зменшення комунікаційної складності застосовується підхід, який полягає у заміні дійсно випадкових значень псевдовипадковими, що генеруються з коротких початкових значень - зерен. Це дозволяє передавати лише компактні зерна замість повних векторів, а всі необхідні величини відновлювати детерміновано за допомогою узгоджених псевдовипадкових генераторів.

Додатково, для підвищення ефективності можуть використовуватись детерміновані зобов'язання замість випадкових. Хоча такий підхід зменшує обсяг переданих даних і спрощує відкриття зобов'язань, він водночас послаблює властивість прихованості, що потребує окремого аналізу з точки зору безпеки.

Маючи хеш-функцію $\mathcal{H}$ можна побудувати схему зобов'язання, яка одночасно забезпечує *прихованість* і *зв'язувальність*. Для цього можна використати $\text{Commit}(s) = (c = \mathcal{H}(s \parallel \rho), St = \rho)$, де ρ — випадкове значення достатньої довжини. Відкриття зобов'язання полягає у розкритті ρ , після чого можна перевірити правильність обчислення. Однак це означає, що кожного разу при розкритті s потрібно також передавати ρ . Такий тип називають *випадковим зобов'язанням*.

Зокрема для схеми підпису Штерна часто використовують інший підхід: $\text{Commit}(s) = (c = \mathcal{H}(s), St = \emptyset)$, де допоміжна інформація відсутня. І для того щоб відкрити значення, достатньо просто показати s , а перевірка полягає у звірці, що $c = \mathcal{H}(s)$. Це зменшує обсяг переданих даних. Проте така схема вже не забезпечує прихованість: зломисник може відрізнити $\text{Commit}(s)$ від $\text{Commit}(s')$ для відомих s і s' , просто обчисливши відповідні геш значення. Такий підхід називають *детермінованим зобов'язанням*.

У роботі [3] наведено один із часто використовуваних методів зменшення обсягу передачі даних в протоколах ідентифікації, в основі якого лежить заміна частини випадкових векторів псевдовипадковими. Використано наступні функції (псевдовипадкові генератори):

$$E_0 : \{0,1\}^{l_{\text{seed}}} \rightarrow \{0,1\}^{l_{\text{seed}}} \times \{0,1\}^{l_{\text{seed}}},$$

$$E_1 : \{0,1\}^{l_{\text{seed}}} \rightarrow \text{Perm}_n,$$

$$E_2 : \{0,1\}^{l_{\text{seed}}} \rightarrow \mathbb{F}_q^n,$$

де $l_{\text{seed}} \in \mathbb{N}$ - розмір зерна.

Зауваження. Так як повна схема складається з R паралельних повторень однораундової схеми, то як зазначають автори у роботі [8], можливо замінити перше повідомлення на глобальне зобов'язання усіх x_i . Це дозволяє розкривати зобов'язання лише для тих значень, які не були безпосередньо розкриті, так як їх можна відновити з відкриттів.

У результаті схема матиме вигляд:

Схема 2.1. Схема ідентифікації Штерна (оптимізована, R раундів), $\text{IS}_{\text{Stern,opt}}$ [8]

Ініціалізація. Нехай $(\mathbf{H}, \mathbf{e}, \mathbf{s} = \mathbf{H}\mathbf{e}) \xleftarrow{\$} \mathcal{D}_{n,k,w}$. Секретний ключ: $sk = \mathbf{e}$, відкритий ключ: $pk = (\mathbf{H}, \mathbf{s})$. Доказувач отримує (pk, sk) , а перевіряльник — pk .

Обмін.

– Для $j = 1, \dots, R$: доказувач обирає випадкове зерно $\text{seed}^j \in \{0,1\}^{l_{\text{seed}}}$ та обчислює $(\text{seed}_{\pi}^j, \text{seed}_y^j) = E_0(\text{seed}^j)$, $\pi^j = E_1(\text{seed}_{\pi}^j)$, $\mathbf{y}^j = (\pi^j)^{-1}(E_2(\text{seed}_y^j))$. Далі він обчислює $\mathbf{t}^j = \mathbf{H}\mathbf{y}^j$, $z_1^j = (\pi^j, \mathbf{t}^j)$, $z_2^j = \pi^j(\mathbf{y}^j)$, $z_3^j = \pi^j(\mathbf{y}^j + \mathbf{e})$, та для $i \in \{1,2,3\}$: $x_i^j = \mathcal{H}_{\text{comm}}(z_i^j)$.

Доказувач надсилає $\tilde{x} = \mathcal{H}_{\text{comm}}(x_1^1, x_2^1, x_3^1, \dots, x_1^R, x_2^R, x_3^R)$ перевіряльнику.

– Для $j = 1, \dots, R$: перевіряльник надсилає випадковий виклик $c^j \xleftarrow{\$} \{1,2,3\}$ доказувачу.

– Для $j = 1, \dots, R$: доказувач розкриває інформацію, щоб перевіряльник міг відновити x_i^j для всіх $i \in \{1,2,3\} \setminus \{c^j\}$. Точніше:

1) якщо $c^j = 1$, доказувач надсилає $l_1^j = (o_2^j, o_3^j)$, де $o_2^j = \text{seed}_y^j$, $o_3^j = \pi^j(\mathbf{e})$, а також x_1^j ;

2) якщо $c^j = 2$, доказувач надсилає $l_2^j = (o_1^j, o_3^j)$, де $o_1^j = \text{seed}_{\pi}^j$, $o_3^j = \mathbf{y}^j + \mathbf{e}$, а також x_2^j ;

3) якщо $c^j = 3$, доказувач надсилає $l_3^j = \text{seed}^j$ та x_3^j .

Перевірка. Для $j = 1, \dots, R$:

1) Якщо $c^j = 1$, перевіряльник відновлює z_2^j, z_3^j з o_2^j, o_3^j , де $z_2^j = E_2(o_2^j)$, $z_3^j = z_2^j + o_3^j$. Далі для $i \in \{2,3\}$ перевіряє, що $\mathcal{H}_{\text{comm}}(x_i^j) = z_i^j$ та що $\text{wt}(o_3^j) = w$.

2) Якщо $c^j = 2$, перевіряльник обчислює $\pi^j = E_1(o_1^j)$, $z_3^j = \pi^j(o_3^j)$, далі $\mathbf{t}^j = \mathbf{H}o_3^j - \mathbf{s}$, $z_1^j = (\pi^j, \mathbf{t}^j)$. Після цього для $i \in \{1,3\}$ перевіряє, що $\mathcal{H}_{\text{comm}}(x_i^j) = z_i^j$.

3) Якщо $c^j = 3$, перевіряльник обчислює

$$(\text{seed}_\pi^j, \text{seed}_y^j) = E_0(l_3^j), \quad \pi^j = E_1(\text{seed}_\pi^j), \quad z_2^j = E_2(\text{seed}_y^j),$$

після чого $\mathbf{t}^j = \mathbf{H}(\pi^j)^{-1}(z_2^j)$, і буде $z_1^j = (\pi^j, \mathbf{t}^j)$. Далі для $i \in \{1, 2\}$ перевіряє, що $\mathcal{H}_{\text{comm}}(x_i^j) = z_i^j$.

Додатково перевіряльник перевіряє, що $\mathcal{H}_{\text{comm}}(x_1^1, x_2^1, x_3^1, \dots, x_1^R, x_2^R, x_3^R)$ відповідає отриманому значенню $\tilde{x}$.

У схемі застосовується паралельне повторення протоколу протягом R раундів, що до суттєво підвищує рівень безпеки за рахунок експоненційного зменшення ймовірності успішного обману. На відміну від однораундової версії, замість передачі окремих комітментів використовується один глобальний комітмент до всіх значень, що зменшує комунікаційні витрати. Крім того, процедура розкриття організована таким чином, що передається лише мінімально необхідна інформація для перевірки, що додатково оптимізує обсяг переданих даних.

На основі цієї схеми ідентифікації, формується схема підпису за допомогою перетворення Фіата–Шаміра. У результаті підпис для повідомлення m має вигляд :

$$\sigma(m) = (\tilde{x}, x_{c_1}^1, \dots, x_{c_R}^R, l_{c_1}^1, \dots, l_{c_R}^R),$$

де вектор викликів визначається як $c = (c_1, \dots, c_R) = \mathcal{H}_{\text{comm}}(\tilde{x} \parallel M)$.

2.2 Атака на схему підпису Штерна з детермінованими зобов'язаннями

Розглянемо атаку на оптимізовану схему підпису Штерна, описану в роботі [8]. Атака використовує можливість виникнення колізій між внутрішніми компонентами різних підписів через застосування детермінованого зобов'язання та зерен обмеженої довжини. Аналіз таких

колізій дозволяє відновити частину внутрішніх параметрів схеми та, зрештою, отримати секретний ключ. Таким чином, дана атака демонструє зниження реального рівня безпеки оптимізованої конструкції порівняно із заявленим.

– Зловмисник виконує $q_s = \lceil \frac{1}{R} 2^{l_{seed}/2} \rceil$ запитів на підпис, не обов'язково для різних повідомлень, щоб отримати підписи $\sigma_{m_1}, \dots, \sigma_{m_{q_s}}$ від підписувача, де кожен підпис $\sigma(m_k)$ має вигляд $\sigma_{m_k} = (\tilde{x}_k, x_{c_{1,k}}^1, \dots, x_{c_{R,k}}^R, l_{c_{1,k}}^1, \dots, l_{c_{R,k}}^R)$.

– Зловмисник намагається знайти пару $((i,k),(i',k'))$ таку, що $(i,k) \neq (i',k') \wedge x_{2,k}^i = x_{2,k'}^{i'} \wedge c_k^i = 2 \wedge c_{k'}^{i'} \neq 2$.

Якщо таких пар не знайдено, повертаються до першого кроку.

– Із такої пари $((i,k),(i',k'))$ зловмисник відновлює секретний ключ наступним чином:

– Оскільки $c_k^i = 2$, то з l_2^i зловмисник може відновити $\pi_k^i = E_1(z_{1,k}^i)$ та $\pi_k^i(y_k^i + e_k) = \pi(z_{3,k}^i)$.

– так як $c_{k'}^{i'} \neq 2$, зловмисник отримує або $seed_{y,k'}^{i'}$, або $seed_{k'}^{i'}$, з якого можна обчислити $seed_{y,k'}^{i'}$.

Крім того, оскільки $x_{2,k'}^i = x_{2,k}^i$, то з переважною ймовірністю $seed_{y,k}^i = seed_{y,k'}^{i'}$, а отже зловмисник отримує $\pi_k^i(y_k^i) = E_2(seed_{y,k'}^{i'})$.

– Маючи π_k^i , $\pi_k^i(y_k^i)$, $\pi_k^i(y_k^i + e)$, зловмисник відновлює секретний ключ, обчислюючи $e = (\pi_k^i)^{-1} (\pi_k^i(y_k^i + e) - \pi_k^i(y_k^i)) = (\pi_k^i)^{-1}(\pi_k^i(e))$, де $(\pi_k^i)^{-1}$ - обернена перестановка до π_k^i .

Теорема 2.1. [8]

Наведена вище атака дозволяє відновити секретний ключ e за час $O(2^{l_{seed}/2})$, виконуючи $O(\frac{1}{r} 2^{l_{seed}/2})$ запитів на підпис. Таким чином, схема забезпечує не більше ніж $\frac{l_{seed}}{2}$ бітів безпеки.

В такому випадку для $\lambda = 128$, існує зловмисник, здатний здійснити атаку шляхом виконання $\frac{1}{R} 2^{64} \leq 2^{58}$ запитів до оракула підпису та обчислень складності порядку 2^{64} . За таких умов схема може бути скомпрометована зі сталою ймовірністю успіху, що оцінюється приблизно як $\frac{1}{10}$.

Водночас обчислювальна складність розв'язання базової задачі декодування синдрому становить близько 2^{128} , і показує суттєву невідповідність між заявленим та фактичним рівнем стійкості. Отже, автори[8] спростували твердження про забезпечення 128-бітного рівня безпеки для даної модифікації.

Аналіз показує, що оптимізація схеми Штерна, що використовує детерміновані коммітменти та зерна фіксованої довжини для генерації випадкових величин допомагає зменшити сам підпис, але при цьому ця модифікація впливає на стійкість схеми так як створюються додаткові залежності між внутрішніми компонентами різних підписів. Зокрема, обмежена ентропія зерен призводить до ненульової ймовірності виникнення колізій між проміжними значеннями, що дозволяє реалізувати атаку використовуючи парадокс «днів народження». За умови накопичення достатньої кількості підписів зломисник отримує можливість встановити зв'язок між окремими внутрішніми параметрами схеми, відновити допоміжні значення й відповідно сам секретний ключ. В підсумку, виходить що фактичний рівень безпеки оптимізованої конструкції є суттєво нижчим за заявлений.

2.3 Усунення вразливості оптимізованої схеми підпису за допомогою рандомізації зерен

Задля підвищення криптографічної стійкості оптимізованої схеми, з детермінованими зобов'язаннями та зернами обмеженої довжини, у роботі [8] було наведено ще один підхід, який має усунути дані вразливості, а саме запропоновано використовувати сіль та індекси.

Основна ідея даної конструкції полягає у використанні випадкового значення $salt \in \{0,1\}^{2\lambda}$, яке генерується під час виконання повної R -раундової схеми ідентифікації та застосовується як додаткове джерело випадковості для всіх зобов'язань і псевдовипадкових генераторів.

Оскільки відповідні функції використовуються багаторазово, до їх вхідних параметрів додається спеціальний індекс, що виконує роль внутрішнього лічильника. Після кожного виклику функції значення індексу змінюється та враховується у вхідних даних, завдяки чому забезпечується незалежність між окремими виконаннями функцій і усувається можливість виникнення небезпечних повторень внутрішніх значень.

Схема 2.2. Оптимізована схема ідентифікації Штерна (R раундів) [8]

Ініціалізація. Нехай $(\mathbf{H}, \mathbf{e}, \mathbf{s} = \mathbf{H}\mathbf{e}) \xleftarrow{\$} \mathcal{D}_{n,k,w}$, де $sk = \mathbf{e}$, $pk = (\mathbf{H}, \mathbf{s})$. Доказувач отримує (pk, sk) , а перевіряльник отримує pk .

Обмін.

Доказувач обирає випадкове значення $salt \in \{0,1\}^{2\lambda}$ та ініціалізує $index = 0$.

Кожного разу при використанні $index$ його значення збільшується.

– Для $j = 1, \dots, R$ доказувач обирає випадкове значення $seed^j \in \{0,1\}^{l_{seed}}$ та визначає $(seed_{\pi}^j, seed_y^j) = E_0(seed^j, salt, index)$, $pi^j = E_1(seed_{\pi}^j, salt, index)$, $y^j = (\pi^j)^{-1}(E_2(seed_y^j, salt, index))$.

Після цього обчислюються $t^j = \mathbf{H}y^j$, $z_1^j = (\pi^j, t^j)$, $z_2^j = \pi^j(y^j)$, $z_3^j = \pi^j(y^j + \mathbf{e})$,

а також для $i \in \{1, 2, 3\}$ обчислюються $x_i^j = \mathcal{H}_{comm}(z_i^j, salt, index)$.

Доказувач надсилає $\tilde{x} = \mathcal{H}_{comm}(x_1^1, x_2^1, x_3^1, \dots, x_1^R, x_2^R, x_3^R, salt, index)$

перевіряльнику разом із $salt$.

– Для $j = 1, \dots, R$ перевіряльник надсилає випадковий виклик $c^j \xleftarrow{\$} \{1, 2, 3\}$.

– Для $j = 1, \dots, R$ доказувач розкриває інформацію, необхідну для відновлення x_i^j , $i \in \{1, 2, 3\} \setminus c$.

Точніше:

1) Якщо $c^j = 1$, доказувач надсилає $l_1^j = (o_2^j, o_3^j)$, де $o_2^j = seed_y$, $o_3^j = \pi^j(\mathbf{e})$, а також x_1^j .

2) Якщо $c^j = 2$, доказувач надсилає $l_2^j = (o_1^j, o_3^j)$, де $o_1^j = seed_\pi$, $o_3^j = y^j + \mathbf{e}$, а також x_2^j .

3) Якщо $c^j = 3$, доказувач надсилає $l_3^j = seed$ та x_3^j .

Перевірка.

Для $j = 1, \dots, R$:

1) Якщо $c^j = 1$, перевіряльник відновлює z_2^j, z_3^j із o_2^j, o_3^j за допомогою $z_2^j = E_2(o_2^j, salt, index)$, $z_3^j = z_2^j + o_3^j$.

Далі для $i \in \{2, 3\}$ перевіряється, що $\mathcal{H}_{comm}(x_i^j, salt, index) = z_i^j$ та $wt(o_3^j) = w$.

2) Якщо $c^j = 2$, обчислюються $\pi^j = E_1(o_1^j, salt, index)$, $z_3^j = \pi^j(o_3^j)$, $t^j = \mathbf{H}o_3^j - s$, та $z_1^j = (\pi^j, t^j)$.

Після цього для $i \in \{1, 3\}$ перевіряється, що $\mathcal{H}_{comm}(x_i^j, salt, index) = z_i^j$.

3) Якщо $c^j = 3$, обчислюються $(seed_\pi^j, seed_y^j) = E_0(l_3^j, salt, index)$, $\pi^j = E_1(seed_\pi^j, salt, index)$, $z_2^j = E_2(seed_y^j, salt, index)$, після чого $t^j = \mathbf{H}(\pi^j)^{-1}(y^j)$.

Тоді будується $z_1^j = (\pi^j, t^j)$. Далі для $i \in \{1, 2\}$ перевіряється, що $\mathcal{H}_{comm}(x_i^j, salt, index) = z_i^j$.

Перевіряльник також перевіряє, що $\mathcal{H}_{comm}(x_1^1, x_2^1, x_3^1, \dots, x_1^R, x_2^R, x_3^R, salt, index)$ дорівнює значенню $\tilde{x}$, яке було отримане від доказувача.

У кожній функції відомо, яке саме значення $index$ використовує доказувач, воно не залежить від внутрішньої випадковості, тому перевіряльник використовує те саме значення $index$ при виконанні всіх перевірок.

2.4 Альтернативні схеми на основі квазіциклічних кодів

У класичній схемі Штерна відкритий ключ задається випадковою матрицею перевірки парності, яка не має спеціальної структури. Через це обсяг пам'яті, необхідний для зберігання параметрів схеми, є дуже великим

навіть для помірних рівнів криптографічної стійкості.

Для розв'язання цієї проблеми у подальших дослідженнях було запропоновано використовувати структуровані коди, зокрема квазіциклічні та *подвійно циркулянтні* коди. Основна ідея такого підходу полягає у заміні випадкових матриць структурованими матрицями, які можна задавати компактним способом без необхідності зберігати всі їх елементи.

Однією з перших робіт у цьому напрямі є стаття [11]

У цій роботі автори запропонували використовувати *подвійно циркулянтні* матриці для оптимізації схеми Штерна та суттєвого зменшення розміру відкритого ключа.

Квазіциклічні та подвійно циркулянтні коди

Квазіциклічні коди є класом лінійних кодів, для яких циклічний зсув кодового слова на фіксовану кількість позицій знову породжує кодове слово цього самого коду.

У випадку *подвійно циркулянтних* конструкції перевірна матриця має вигляд

$$H = (I \mid A),$$

де: I - одинична матриця, A - циркулянтна матриця.

Циркулянтна матриця повністю визначається своїм першим рядком:

$$A = \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & a_l \\ a_l & a_1 & a_2 & \cdots & a_{l-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_2 & a_3 & a_4 & \cdots & a_1 \end{pmatrix}.$$

Кожний наступний рядок отримується циклічним зсувом

попереднього рядка. Завдяки цьому для зберігання всієї матриці достатньо лише одного рядка довжини l .

У класичній схемі Штерна необхідно зберігати всю випадкову матрицю розміру

$$(n - k) \times n,$$

тоді як у *подвійно циркулянтному* випадку достатньо лише:

- першого рядка матриці A ;
- синдрому.

Таким чином, розмір відкритого ключа зменшується майже в декілька разів.

2.5 Схеми ідентифікації Верона

Паскаль Верон опублікував роботу [19] в 1996 році, у якій запропонував дуальну версію схеми Штерна, стійкість якої ґрунтується на задачі загального декодування (General Decoding, GD).

Задача загального декодування (GD). [3]

Нехай задано $G \xleftarrow{\$} \mathbb{F}_2^{k \times n}$, $y \xleftarrow{\$} \mathbb{F}_2^n$, та ціле число $\omega > 0$.

Задача полягає у знаходженні пари $(m, e) \in \mathbb{F}_2^k \times \mathbb{F}_2^n$, такої, що $wt(e) \leq \omega$ та $mG \oplus e = y$.

Зауважимо, що $x = mG \in \mathbb{F}_2^n$ для $m \in \mathbb{F}_2^k$ за означенням є кодовим словом. Іншими словами, задача GD полягає у тому, що для заданого вектора $y \in \mathbb{F}_2^n$ необхідно знайти (єдине) кодове слово $x \in C$, для якого значення $wt(x - y)$ є мінімальним.

Схема Верона використовує породжувальну матрицю коду замість перевіркової матриці, що дозволяє дещо зменшити комунікаційні витрати під час виконання протоколу. Як і схема Штерна, вона належить до багаторазових протоколів з нульовим розголошенням, у яких кожен раунд реалізується у вигляді трипрохідної взаємодії між доказувачем і перевіряльником. Імовірність успішного обману в межах одного раунду

становить $\frac{2}{3}$.

Схема 2.3. Схема ідентифікації Верона[19]

Генерація ключів (KeyGen).

Нехай κ — параметр безпеки.

Обираються параметри n, k, ω таким чином, щоб $WF_{ISD}(n, k, \omega, 2) \geq 2^\kappa$.

Далі випадковим чином генерується породжувальна матриця $G \leftarrow \mathbb{F}_2^{k \times n}$.

Після цього обирається пара $(m, e) \leftarrow \mathbb{F}_2^k \times \mathbb{F}_2^n$, така, що $wt(e) = \omega$, де (m, e) є секретним ключем.

Далі обчислюється $y \leftarrow mG \oplus e$, де y є відкритим ключем.

У результаті отримуємо: $(sk, pk) = ((m, e), (y, G, \omega))$.

Обмін

– Перевіряльник випадково обчислює:

- двійковий вектор u довжини k ;
- випадкову перестановку σ множини $\{1, \dots, n\}$;

і надсилає B три комітменти: $c_1 = \langle \sigma \rangle$, $c_2 = \langle (u + m)G\sigma \rangle$, $c_3 = \langle (uG + x)\sigma \rangle$.

– B надсилає випадковий елемент $b \in \{0, 1, 2\}$.

– Якщо $b = 0$:

- A розкриває $u + m$ та σ ;
- B перевіряє коректність c_1 та c_2 (оскільки G та $\langle \cdot \rangle$ є відкритими).

– Якщо $b = 1$:

- A розкриває $(u + m)G\sigma$ та $e\sigma$;
- B перевіряє коректність c_2 та c_3 , а також те, що $\omega(e\sigma) = p$.

Оскільки $x = mG + e$, то $(u + m)G\sigma + e\sigma = (uG + x)\sigma$.

– Якщо $b = 2$:

- A розкриває σ та u ;
- B перевіряє коректність c_1 та c_3 (оскільки $\langle \cdot \rangle$, G та x є відкритими).

В загальному, задачі SD та GSD є еквівалентними і відрізняються лише способом представлення базового коду: у першому випадку

використовується перевірна матриця, а в другому - породжувальна матриця. Обидва протоколи потребують виконання трьох раундів і мають ймовірність успішного обману $\frac{2}{3}$.

2.6 Модифікація протоколу Верона до рангової метрики на основі подвійних циркулянтних кодів

Схема RVDC (Rank Veron Double Circulant), досліджена в роботі [1] ґрунтується на спеціальному класі лінійних кодів - подвійно циркулянтних кодах.

Означення 2.1. Нехай $n = 2k$ для деякого цілого k . Лінійний $[n, k]$ -код C над полем $\mathbb{F}_q$ є *подвійно циркулянтним* (DC-кодом), якщо для будь-якого кодового слова $c = (c_1, c_2) \in C$ одночасний лівий циклічний зсув обох блоків також є кодовим словом.

Систематичний DC-код має перевірочну матрицю вигляду $H = [I_k \mid A]$, де: I_k - одинична матриця розміру k ; A - циклічна матриця розміру $k \times k$.

Подвійно циркулянтні коди є підкласом квазіциклічних кодів і дозволяють суттєво скоротити розмір публічного ключа: для зберігання всієї матриці достатньо одного рядка.

Нехай зафіксовано базис $b = \{b_1, \dots, b_m\}$ поля $\mathbb{F}_{q^m}$ над $\mathbb{F}_q$. Тоді будь-який вектор $a \in \mathbb{F}_{q^m}^n$ може бути представлений матрицею над $\mathbb{F}_q$ розміру $m \times n$ шляхом розкладання кожної компоненти по базису b . Ранг вектора a визначається як ранг цієї матриці.

Нехай $Q \in M_{m,m}(\mathbb{F}_q)$, $P \in M_{n,n}(\mathbb{F}_q)$, $v \in \mathbb{F}_{q^m}^n$. Визначимо функцію $\Pi_{P,Q}(v) = \varphi^{-1}(Q \cdot \varphi(v) \cdot P)$, де φ - матричне представлення вектора по базису b . Ця функція є ранговим аналогом перестановки: вона зберігає рангову вагу вектора $w_R(\Pi_{P,Q}(v)) = w_R(v)$ та є лінійною.

Для будь-яких двох векторів однакового рангу існують матриці P та Q , що переводять один у другий - аналогічно до того, як будь-яка

перестановка переводить два вектори однакової ваги Геммінга один в один.

RSD-розподіл

Для натуральних чисел n, k, r розподіл $RSD(n, k, r)$ обирає $H \leftarrow \mathbb{F}_{q^m}^{(n-k) \times n}$ та $x \leftarrow \mathbb{F}_{q^m}^n$ таким чином, що $w_R(x) = r$, і повертає пару $(H, H \cdot x^T)$.

RSD-задача

На вхід подається пара (H, y^T) з RSD-розподілу.

Потрібно знайти $x \in \mathbb{F}_{q^m}^n$ таке, що $H \cdot x^T = y^T$ та $w_R(x) = r$.

Означення 2.2. Для вектора $y = (y_1, y_2) \in \mathbb{F}_{q^m}^n$, де $n = 2k$, паралельним лівим зсувом на i позицій називається операція $n = 2k$, що одночасно циклічно зсуває обидві половини y_1 та y_2 на i позицій.

Для вектора $a = (\alpha_1, \dots, \alpha_k) \in \mathbb{F}_q^k$ визначаються:

$$\Gamma_a(y) = \sum_{i=1}^k \alpha_i \cdot \text{drot}_i(y) \in \mathbb{F}_{q^m}^n,$$

$$\Gamma'_a(x) = \sum_{i=1}^k \alpha_i \cdot \text{rot}_{k-i}(x) \in \mathbb{F}_{q^m}^k.$$

Лема 2.1. Для породжувальної матриці G подвійно циркулянтного коду та вектора $x \in \mathbb{F}_{q^m}^k$ виконується: $\Gamma_a(x \cdot G) = \Gamma'_a(x) \cdot G$.

Дана лема є важливою для доведення коректності (*completeness*) протоколу: вона гарантує, що перевіряльник може перевірити цілісність відповіді доказувача без знання секрету.

Далі описано схему RVDC[1]

Схема 2.4. Схема ідентифікації (5-прохідна)

Алгоритм генерації ключів $KGen$

- обираються параметри безпеки m, n, k, r
- генерується випадковий вектор $x \leftarrow \mathbb{F}_{q^m}^k$;
- генерується вектор помилки $e \leftarrow \mathbb{F}_{q^m}^n$ таким чином, що $w_R(e) = r$;
- секретний ключ визначається як $sk = (x, e)$;

- генерується породжувальна матриця $G \leftarrow \mathbb{F}_{q^m}^n$ яка розширюється до подвійно циркулянтної форми G' ;
- обчислюється відкритий вектор $y \leftarrow x \cdot G' \oplus e$, - приховане кодове слово;
- публічний ключ визначається як $pk = (y, G, r)$.

Структура ключів відображає ту саму ідею, що й у класичному протоколі Верона: секрет - це пара (x, e) , де x є повідомленням-рядком, а e - вектором помилки; публічний ключ y є прихованим кодовим словом.

Захищеність ґрунтується на неможливості ефективно знайти будь-яку пару (x', e') таку, що $x' \cdot G' \oplus e' = y$ та $w_R(e') = r$ без знання секретного ключа.

Обмін

Нехай доказувач хоче довести свою особу перевіряльнику та знає лише $pk = (y, G, r)$.

Протокол складається з R раундів, кожен з яких виконується за наступною 5-прохідною схемою:

Прохід 1 (перше зобов'язання).

Доказувач генерує маскувальні параметри:

- випадковий вектор $u \leftarrow \mathbb{F}_{q^m}^k$;
- матриці $Q \leftarrow M_{m,m}(\mathbb{F}_q)$ та $P \leftarrow M_{n,n}(\mathbb{F}_q)$, що задають функцію $\Pi_{P,Q}$.

Обчислюються два зобов'язання та надсилаються перевіряльнику: $c_1 \leftarrow H(P, Q)$, $c_2 \leftarrow H(\Pi_{P,Q}(u \cdot G))$.

Прохід 2 (перший виклик).

Перевіряльник надсилає рандомізований challenge $a = (\alpha_1, \dots, \alpha_k) \leftarrow \mathbb{F}_q^k$, де не всі α_i однакові.

Прохід 3 (третє зобов'язання).

Доказувач обчислює: $c_3 \leftarrow H(\Pi_{P,Q}(u \cdot G + \Gamma_a(e)))$

і надсилає c_3 перевіряльнику.

Прохід 4 (другий виклик). Перевіряльник надсилає двійковий виклик $b \leftarrow \{0,1\}$.

Прохід 5 (відповідь). Доказувач відкриває інформацію залежно від b :

Якщо $b = 0$: надсилаються $r_1 = (P, Q)$ та $r_2 = u + \Gamma'_a(x)$.

Перевіряльник перевіряє: $c_1 = H(r_1)$ та $c_3 = H(\Pi_{\{r_1\}}(r_2 \cdot G + \Gamma_a(y)))$.

Якщо $b = 1$: надсилаються $r_1 = \Pi_{\{P, Q\}}(u \cdot G)$ та $r_2 = \Pi_{\{P, Q\}}(\Gamma_a(e))$.

Перевіряльник перевіряє: $c_2 = H(r_1)$, $c_3 = H(r_1 + r_2)$ та $w_R(r_2) = r$.

Логіка протоколу полягає в тому, що за виклику $b = 0$ доказувач розкриває маскування компоненти x (через $u + \Gamma'_a(x)$) і матрицю $\Pi_{\{P, Q\}}$, але не розкриває e ; за виклику $b = 1$ - розкриває замасковані варіації uG та $\Gamma_a(e)$, але не розкриває x . Таким чином, незалежно від виклику, злоумисник не отримує обох компонент секрету одночасно.

Процес перетворення на схему підпису (сRVDC)

Схема ідентифікації RVDC перетворюється на схему цифрового підпису сRVDC (*compressed RVDC*) за допомогою узагальнення перетворення Фіата–Шаміра для n -прохідних протоколів.

Підпис повідомлення M у схемі сRVDC з R раундами має вигляд:

$$\sigma(M) = (\widetilde{\{x\}}, x_{\{c_1\}}^1, \dots, x_{\{c_R\}}^R, l_{\{c_1\}}^1, \dots, l_{\{c_R\}}^R),$$

де вектор викликів $(c_1, \dots, c_R) = H(\widetilde{\{x\}} \parallel M)$ є геш-значенням від глобально зобов'язання та повідомлення.

Глобальне зобов'язання

$$\widetilde{\{x\}} = H(x_1^1, x_2^1, x_3^1, \dots, x_1^R, x_2^R, x_3^R)$$

замінює окремі зобов'язання всіх раундів, що дозволяє зменшити обсяг підпису.

Незважаючи на компактність рангового представлення, схема RVDC не є криптографічно стійкою в поточному вигляді. У роботі [17] було доведено, що вона не задовольняє властивість повноти і може бути

зламана атакою Rank Support Learning [9]. Таким чином, схема потребує суттєвого доопрацювання перед практичним застосуванням, зокрема перегляду механізму формування першого виклику протоколу з метою зміни носія секретного вектора на кожному раунді.

Висновки до розділу 2

У другому розділі розглянуто оптимізації комунікаційної складності протоколів ідентифікації, засновані на використанні псевдовипадкових генераторів та детермінованих зобов'язань. Показано, що застосування детермінованих зобов'язань із зернами фіксованої довжини ℓ_{seed} знижує рівень безпеки оптимізованої схеми підпису Штерна до $\ell_{\text{seed}}/2$ біт замість заявлених ℓ_{seed} біт. Описано атаку [8], що реалізує це зниження через колізії псевдовипадкових зерен за парадоксом днів народження, та наведено метод її усунення через використання солі та індексу. Також розглянуто схеми на основі квазіциклічних та подвійних циркулянтних кодів, схему Верона як дуальну версію протоколу Штерна, та її узагальнення у ранговій метриці у вигляді схеми Rank Veron Double Circulant.

3 ВИКОРИСТАННЯ СХЕМ ШТЕРНА І ВЕРОНА ДЛЯ ПРИСТРОЇВ З ОБМЕЖЕНИМИ РЕСУРСАМИ

В цьому розділі побудовано оптимізацію схеми цифрового підпису Штерна на схему Верона та досліджено вплив параметрів схем на можливість застосування на малоресурсних пристроях.

3.1 Побудова модифікації схеми цифрового підпису Верона

У розділі 2.3 було представлено оптимізацію схеми підпису Штерна, з роботи [8] через використання солі та індексів.

Так як у даному розділі буде проаналізовано використання схем цифрового підпису Штерна і Верона на малоресурсних пристроях, то постає необхідність модифікувати й схему Верона для зменшення розміру підпису і ключів.

Конструкція, запропонована в роботі [8] з використанням солі й індексу рандомізує всі виклики геш-функції та псевдовипадкових генераторів спільними значеннями $salt \in \{0,1\}^{2\lambda}$ та внутрішнім лічильником $index$ і так як вона стосується лише способу виведення випадковості та обчислення зобов'язань, а не алгебраїчної структури схеми то її можна перенести на схему Верона.

З попереднього розділу відомо, що схема Верона на відмінну від Штерна у схемі Верона використовує породжувальну матриця лінійного коду $G \in \mathbb{F}_2^{k \times n}$.

У спрощеному вигляді відкритий вектор можна подати як $x = uG + e$, де $u \in \mathbb{F}_2^k$ є двійковим вектором, uG є кодовим словом, а $e \in \mathbb{F}_2^n$, $wt(e) = w$, є вектором помилки малої ваги. Тоді відкритий ключ схеми Верона можна подати у вигляді $pk = (G, x)$.

У цій схемі доказувач доводить знання таких значень u та e , для яких

$$x = uG + e.$$

Схема 3.1. Оптимізована схема Верона

Ініціалізація. Відбувається аналогічно як у і класичній схемі Верона

Обмін. Доказувач обирає $salt \in \{0,1\}^{2\lambda}$ та ініціалізує $index = 0$; кожен виклик функції збільшує $index$.

– Для j від 1 до r : доказувач обирає випадкове значення $seed^j \in \{0,1\}^{\ell_{seed}}$ і визначає $(seed_{\sigma}^j, seed_u^j) = E_0(seed^j, salt, index)$, $\sigma^j = E_1(seed_{\sigma}^j, salt, index)$, $w^j = E_2(seed_u^j, salt, index)$.

Обчислює

$$c_1^j = \langle \sigma^j, salt, index \rangle, c_2^j = \langle (u^j + m)G\sigma^j, salt, index \rangle, c_3^j = \langle (u^j G + y)\sigma^j, salt, index \rangle.$$

Доказувач надсилає $\tilde{c} = \mathcal{H}_{comm}(c_1^1, \dots, c_3^r, salt, index)$ та $salt$.

– Для j від 1 до r : Перевіряльник надсилає виклик $b^j \in \{0,1,2\}$.

– Доказувач розкриває інформацію:

$b^j = 0$: доказувач надсилає $seed_{\sigma}^j$ та $w^j + m$; перевіряльник перевіряє c_1^j, c_2^j ;

$b^j = 1$: $(u^j + m)G\sigma^j$ та $e\sigma^j$; Перевіряльник перевіряє c_2^j, c_3^j і $wt(e\sigma^j) = \omega$;

$b^j = 2$: $seed^j$; перевіряльник перевіряє c_1^j, c_3^j .

Перевірка. У гілці $b^j = 1$ використовується тотожність $(u^j + m)G\sigma^j \oplus e\sigma^j = (u^j G + y)\sigma^j$ (з $y = mG \oplus e$). Наприкінці перевіряльник перевіряє, що $\mathcal{H}_{comm}(c_1^1, \dots, c_3^r, salt, index) = \tilde{c}$.

Для побудованої модифікації схеми Верона доведемо коректність та стійкість.

Твердження 3.1. Якщо підписувач має правильний секретний ключ (m, e) , то сформований ним підпис приймається з ймовірністю 1.

Доведення. Розглянемо довільний раунд j і покажемо, що в кожній з трьох гілок перевіряльник відновлює саме ті значення, які було зафіксовано в зобов'язаннях.

Гілка $b^j = 0$. Перевіряльник отримує $seed_{\sigma}^j$ та $w^j + m$. З $seed_{\sigma}^j$ він відновлює перестановку σ^j і перевіряє зобов'язання c_1^j , а з $w^j + m$ обчислює

$(u^j + m)G\sigma^j$ і перевіряє c_2^j . Обидва значення збігаються з обчисленими чесним підписувачем.

Гілка $b^j = 1$. Перевіряльник отримує $(u^j + m)G\sigma^j$ та $e\sigma^j$. Зобов'язання c_2^j перевіряється безпосередньо, а c_3^j - через тотожність $(u^j + m)G\sigma^j \oplus e\sigma^j = (u^jG + mG \oplus e)\sigma^j = (u^jG + y)\sigma^j$, яка виконується, так як $mG \oplus e = y$. Перевірка ваги $wt(e\sigma^j) = \omega$ виконується, бо перестановка зберігає вагу Геммінга: $wt(e\sigma^j) = wt(e) = \omega$.

Гілка $b^j = 2$. Перевіряльник отримує $seed^j$, з якого відновлює σ^j та u^j , і перевіряє зобов'язання c_1^j та c_3^j .

У кожній гілці всі перевірки чесного підписувача проходять детерміновано, тому відновлене глобальне зобов'язання $\tilde{c}$ збігається з наданим, і підпис приймається з імовірністю 1. $\square$

Твердження 3.2. *Ймовірність того, що злоумисник без знання секретного ключа сформує підпис, який буде прийнято, не перевищує $(\frac{2}{3})^R$.*

Доведення. Розглянемо один раунд. Злоумисник, що не знає (m, e) , фіксує зобов'язання c_1^j, c_2^j, c_3^j до отримання виклику. Покажемо, що той, хто здатний вірно відповісти на всі три виклики $b \in \{0, 1, 2\}$ того самого раунду, фактично розв'язує задачу загального декодування. З відповідей на виклики $b = 0$ та $b = 2$ відновлюються σ^j , u^j та $u^j + m$, а отже вектор m ; з відповіді на виклик $b = 1$ відновлюється вектор $e\sigma^j$ ваги ω , а отже вектор e . Таким чином, злоумисник, який не має секрету, здатний підготувати зобов'язання, що дають коректну відповідь щонайбільше на два з трьох викликів, тому ймовірність успішного обману в одному раунді не перевищує $\frac{2}{3}$. Оскільки виклики раундів незалежні, ймовірність успішного обману в усіх R раундах дорівнює $(\frac{2}{3})^R$.

Стійкість до відновлення секретного ключа ґрунтується на тому, що знаходження пари (m, e) за відкритим ключем $y = mG \oplus e$ з $wt(e) = \omega$ є задачею загального декодування. Ця модифікація не знижує рівень стійкості, так як у ній лише забезпечується формування різних вхідних

3.2 Вимоги до постквантового підпису для IoT

На сьогодні малоресурсні IoT-пристрої досить стрімко інтегруються в критичні сфери: медицину, промислову автоматизацію та фінансові сервіси і тому виникає потреба в забезпеченні автентифікації джерела даних і контролю їх цілісності відповідно до цього використовується цифрового підпису. Але за рахунок того, що вони мають обмежені обчислювальні ресурси, обсяг пам'яті та пропускну здатність каналів зв'язку накладають додаткові вимоги на схеми підпису.

Багато IoT-пристроїв розгортаються на тривалий термін і не можуть бути легко оновлені. Наприклад, оновлення захисту кардіостимулятора потребує хірургічного втручання, а медичні журнали мають залишатись верифікованими протягом десятиліть. Класичні схеми на основі факторизації та дискретного логарифма є вразливими до квантових алгоритмів, що робить перехід до постквантової криптографії невідкладним.

Для пристроїв з батарейним живленням обчислювальна складність підпису безпосередньо впливає на термін служби батареї і може стати критичним чинником для імплантованих медичних пристроїв, де заміна батареї потребує хірургічного втручання.

Стандартизовані NIST схеми вимагають значно більше ресурсів: для ML-DSA сукупний обсяг відкритого, секретного ключів і підпису становить близько 6,2 КБ, а підпис SLH-DSA - від $\approx 7,7$ КБ до ≈ 49 КБ [14, 15], що не підходить для пристроїв із жорсткими обмеженнями пам'яті та каналу передачі.

Стійкість схем підпису для IoT мають ґрунтуватись на добре досліджених задачах та забезпечувати стійкість до атак побічних каналів, яка є особливо важливою для вбудованих систем.

Штерн-подібні схеми підпису задовільняють частину вимог: вони є

постквантово стійкими і їх стійкість ґрунтується на добре дослідженій задачі з простими бітовими операціями- XOR, зсувами та гешуванням і відповідно це спрощує реалізацію та захист від атак побічними каналами. Завдяки малим ключам, схеми з застосуванням квазіциклічної оптимізації задовільняють і вимоги до пам'яті, але виникає проблема з каналом передачі: великий розмір підпису впливає на ефективність його передачі бездротовими каналами зв'язку.

3.3 Вплив параметрів схем Штерна та Верона на ресурси пристроїв

Схеми Штерна та Верона характеризуються трьома основними параметрами: довжиною коду n , розмірністю k та кількістю раундів R . Ці параметри пов'язані з рівнем безпеки λ і безпосередньо визначають вимоги до флеш-пам'яті, оперативної пам'яті та часу передачі його каналом зв'язку.

Як вже було зазначено в попередніх розділах, ймовірність обману зломисником в одному раунді становить q . Відповідно щоб зменшити ймовірність до нехтовно малої, протокол виконується незалежно R разів, унаслідок цього ймовірність обману дорівнює q^R .

З умови стійкості $q^R \leq 2^{-\lambda}$, отримуємо:

$$R = \left\lceil \frac{\lambda}{\log_2 \frac{1}{q}} \right\rceil. \quad (3.1)$$

Для 3-прохідних схем Штерна та Верона $q = 2/3$, тому $R \approx 1,71\lambda$, отже для рівня безпеки 128 кількість раундів становить 219.

Параметр n визначає вимогою стійкості до атак ISD [7] залежно від λ . Зі схеми Верона [19] знаємо, що при $n = 512$, $k = 256$, $w = 56$ складність атаки ISD становить $\approx 2^{70}$. Мінімальне значення n , що забезпечує рівень безпеки $\lambda = 128$ біт, визначається чисельно за

формулою складності ISD [16]:

Реалізований програмний пошук дає мінімальне $n = 978$. Беремо $n = 1024 = 2^{10}$ так як це найближчий степінь двійки.

Найбільшу частину флеш-пам'яті у схемах Штерна займає відкритий ключ - матриця перевірки парності H , яка у класичній схемі є випадковою матрицею розміру $(n - k) \times n$ біт.

При $n = 1024$, $k = 512$:

$$|H|_{\text{без QC}} = \frac{(n - k) n}{8} = \frac{512 \cdot 1024}{8} = 65\,536 \text{ байт} = 64 \text{ КБ.}$$

Для прикладу можна розглянути мікроконтролер STM32F030C8T6 (ядро Cortex-M0, 64 КБ флеш-пам'яті, 8 КБ оперативної), так як він один із найбільш обмежених і найдешевших 32-бітних мікроконтролерів.

Після розміщення програмного коду (≈ 20 КБ [6]) залишається лише 44 КБ, що є меншим за розмір матриці. Отже, без оптимізації схема непридатна для цього класу пристроїв через нестачу флеш-пам'яті.

В схемі з квазіциклічною оптимізацією [11] матриця перевірки парності замінюється на матрицю $H = (I | A)$, де A - циркулянтна матриця, яка повністю визначається своїм першим рядком довжиною $n/2 = 512$ біт. Разом із синдромом розмір відкритого ключа зменшується до: $n/8 = 128$ Б

Із приватним ключем ($|sk| = n/8 = 128$ байт) загальний розмір ключів становить 256 байт - менше 0,5% флеш-пам'яті для STM32F030C8T6.

Цей результат можемо описати таким твердженням.

Твердження 3.3. *Для будь-якого фіксованого пристрою з флеш-пам'яттю F байтів існує порогове значення $\lambda_0(F)$ таке, що при $\lambda > \lambda_0(F)$ схема Штерна без квазіциклічної оптимізації є непридатною через нестачу флеш-пам'яті.*

Розглядаючи схему Штерна, підпис складається з глобального зобов'язання (2λ біт), значення salt (2λ біт), R розкритих зобов'язань (по

2λ біт) та R відповідей.

Розмір відповіді залежить від виклику $c \in \{1,2,3\}$, тому розглянемо усереднену по трьох рівноймовірних викликах оцінку. У викликах $c = 1$ та $c = 2$ доказувач розкриває одне зерно (ℓ_{seed} біт) та один вектор довжини n , тоді як у виклику $c = 3$ - лише зерно.

Середній розмір відповіді одного раунду для схеми Штерна:

$$|l| = \frac{(\ell_{\text{seed}} + n) + (\ell_{\text{seed}} + n) + \ell_{\text{seed}}}{3} = \ell_{\text{seed}} + \frac{2n}{3},$$

де $\ell_{\text{seed}} = \lambda$ (використовуючи оптимізацію з сіллю та індексом) [8]).

Тоді отримуємо повний розмір підпису Штерна:

$$|\sigma| = 4\lambda + R \left(2\lambda + \ell_{\text{seed}} + \frac{2n}{3} \right). \quad (3.2)$$

Якщо $R = 1,71\lambda$, $\ell_{\text{seed}} = \lambda$, $n = 8\lambda$, то отримаємо:

$$|\sigma| \approx 4\lambda + 1,71\lambda \left(2\lambda + \lambda + \frac{16\lambda}{3} \right) = 4\lambda + 1,71\lambda \cdot \frac{25\lambda}{3} \approx 14,3 \lambda^2.$$

При $\lambda = 128$: $14,3 \cdot 128^2 = 14,3 \cdot 16\,384 \approx 234\,291$ біт $\approx 28,6$ КБ

$$|\sigma|_{\text{Stern}} \approx 28,6 \text{ КБ};$$

Аналогічно можемо обчислити для схеми Верона:

Обсяги розкритих відповідей за викликами мають такий вигляд: при $b = 0$ передається $\ell_{\text{seed}} + k$ біт (seed_σ та $u + m$); при $b = 1$ передається $2n$ біт (два переставлені вектори $(u + m)G\sigma$ та $e\sigma$); при $b = 2$ передається ℓ_{seed} біт, лише зерно. Середній розмір відповіді одного раунду дорівнює

$$|l| = \frac{(\ell_{\text{seed}} + k) + 2n + \ell_{\text{seed}}}{3} = \frac{2\ell_{\text{seed}} + k + 2n}{3}. \quad (3.3)$$

Повний розмір підпису складається із солі довжини 2λ , глобального комітменту довжини 2λ , а також даних кожного раунду: одного надісланого комітменту довжини 2λ та відповіді розміру $|l|$. Отже,

повний розмір підпису для схеми Верона можна оцінити як

$$|\sigma| = 4\lambda + r \left(2\lambda + \frac{2\ell_{\text{seed}} + k + 2n}{3} \right). \quad (3.4)$$

отримуємо, що $|l| = \frac{2 \cdot 128 + 512 + 2 \cdot 1024}{3} = \frac{2816}{3} \approx 939$ біт.

Тоді

$$|\sigma| = 4 \cdot 128 + 219 \left(2 \cdot 128 + \frac{2 \cdot 128 + 512 + 2 \cdot 1024}{3} \right) \approx 32,0 \text{ КБ.}$$

Як приклади для оцінювання характеристик передачі підпису обрано протоколи BLE5.0 та Zigbee, так як вони часто використовуються в IoT пристроях і вбудованих системах з обмеженими ресурсами. BLE має вищу швидкість передачі даних і часто застосовується в носимих, медичних та мобільних пристроях. Натомість Zigbee забезпечує підтримку мереж із великою кількістю вузлів та використовується в системах автоматизації й технологіях «розумного дому».

Так як MTU бездротових протоколів обмежений, то підпис треба фрагментувати.

Відповідно можна визначити кількість пакетів : $N = \left\lceil \frac{|\sigma|}{M-h} \right\rceil$

та час передачі: $T \approx N \cdot \frac{M}{v}$,

де M - MTU, h - розмір заголовку, v - швидкість каналу.

Зауваження. Отримане значення є наближеною оцінкою часу передачі, оскільки останній пакет може бути заповнений не повністю.

Параметри:

BLE 5.0 ($M = 251$, $h = 4$ байти, $v = 1$ Мбіт/с),

Zigbee ($M = 127$, $h = 25$ байт, $v = 250$ Кбіт/с).

Розрахуємо для схеми Штерна через BLE: $N = \left\lceil \frac{29 \cdot 286}{247} \right\rceil = 119$ пакетів;

$$T = \frac{119 \cdot 251}{125000} = 239 \text{ мс.}$$

Результати для розглянутих схем наведено у Таблиці 3.1.

Числові параметри взято зі специфікацій відповідних протоколів.

Для BLE 5.0 [4] максимальне навантаження канального рівня після розширення довжини даних становить $M = 251$ байт при заголовку $h \approx 4$

Схема	R	$ \sigma $	$ sk + pk $	N_{BLE}	T_{BLE}	T_{Zigbee}
Штерн	219	28,6 КБ	256 Б	119	239 мс	1170 мс
Верон	219	32,0 КБ	384 Б	100	201 мс	979 мс

Таблиця 3.1 – Характеристики схем при $\lambda = 128$ біт

байти та швидкості базового фізичного рівня $v = 1$ Мбіт/с.

Для Zigbee на основі IEEE 802.15.4 [12] максимальний кадр обмежено $M = 127$ байтами, службовий заголовок MAC-рівня з адресацією становить $h \approx 25$ байт, а швидкість фізичного рівня O-QPSK у діапазоні 2,4 ГГц - $v = 250$ Кбіт/с.

Висновки до розділу 3

У цьому розділі побудовано модифікацію схеми Верона, за використання спільного значення солі та індексу для формування вхідних даних геш-функцій і псевдовипадкових генераторів. Такий підхід дозволяє зменшити обсяг даних, які необхідно зберігати або передавати під час формування підпису. Відповідно до цієї модифікації було доведено твердження стійкості та коректності для схеми. Також було досліджено вплив параметрів схем Штерна і Верона на можливість їх реалізації на малоресурсних пристроях, зокрема розміри ключів, підписів з урахуванням модифікацій. Аналітична оцінка показала, що розміри підписів із модифікованими схемами є меншими в декілька разів, порівняно з класичними схемами.

ВИСНОВКИ

У результаті роботи досліджено застосування схем цифрового підпису Штерна і Верона в малоресурсних пристроях. Отримано результати, що дозволяють оцінити доцільність використання цих схем у пристроях з обмеженими обчислювальними ресурсами.

Розглянуто принципи побудови схеми підпису Штерна та наявні оптимізації, спрямовані на зменшення обсягу підпису, зокрема квазіциклічні коди, використання детермінованих зобов'язань, солі та індексів.

У роботі побудовано модифікацію схеми Верона, за використання спільного значення солі та індексу для формування вхідних даних ген-функцій і псевдовипадкових генераторів. Такий підхід дозволяє зменшити обсяг даних, які необхідно зберігати або передавати під час формування підпису. Відповідно до цієї модифікації було доведено твердження стійкості та коректності.

Виконано порівняльний аналіз оптимізованих схем Штерна і Верона за розмірами відкритих ключів, секретних ключів і підписів. Також побудовано оцінку ресурсної складності застосування схем Штерна і Верона в малоресурсних пристроях. Показано, що за квазіциклічної оптимізації та використання оптимізації з сіллю та індексом, обсяг ключів обох схем зменшується до сотень байтів (256 Б для схеми Штерна та 384 Б для схеми Верона), а розміри підпису на рівні стійкості 128 біт становлять близько 28–32 КБ.

В підсумку, схеми цифрового підпису Штерна і Верона є перспективними з погляду постквантової стійкості, однак їх використання в малоресурсних пристроях потребує додаткової оптимізації. Подальші дослідження можуть стосуватись аналізу стійкості оптимізованої схеми Верона до атак ISD та практичне моделювання використання пам'яті й часу виконання на реальних малоресурсних пристроях.

ПЕРЕЛІК ПОСИЛАНЬ

- [1] Emanuele Bellini та ін. «Improved Veron Identification and Signature Schemes in the Rank Metric». В: *CoRR* abs/1903.10212 (2019). arXiv: 1903.10212. URL: <http://arxiv.org/abs/1903.10212>.
- [2] Elwyn R. Berlekamp, Robert J. McEliece та Henk C. A. van Tilborg. «On the inherent intractability of certain coding problems». В: *IEEE Transactions on Information Theory* 24.3 (1978), с. 384—386.
- [3] Loïc Bidoux та ін. «Quasi-Cyclic Stern Proof of Knowledge». В: (2022). arXiv: 2110.05005 [cs.CR]. URL: <https://arxiv.org/abs/2110.05005>.
- [4] Bluetooth SIG. *Bluetooth Core Specification, Version 5.0*. Тех. звіт. Version 5.0. Bluetooth Special Interest Group, 2016. URL: <https://bluetooth.com>.
- [5] Dan Boneh та Victor Shoup. *A Graduate Course in Applied Cryptography*. 2015. URL: https://crypto.stanford.edu/~dabo/cryptobook/draft_0_2.pdf.
- [6] Pierre-Louis Cayrel, Philippe Gaborit та Emmanuel Prouff. «Secure Implementation of the Stern Authentication and Signature Schemes for Low-Resource Devices». В: *Lecture Notes in Computer Science* (2008). За ред. Gilles Grimaud та François-Xavier Standaert, с. 191—205. DOI: 10.1007/978-3-540-85893-5_14. URL: https://doi.org/10.1007/978-3-540-85893-5_14.
- [7] Florent Chabaud. «On the Security of Some Cryptosystems Based on Error-Correcting Codes». В: (1994).
- [8] André Chailloux та Simona Etinski. «On the (in)security of optimized Stern-like signature schemes». В: *Designs, Codes and Cryptography* 92.3 (груд. 2023), с. 803—832. ISSN: 1573-7586. DOI: 10.1007/s10623-023-01329-y. URL: <http://dx.doi.org/10.1007/S10623-023-01329-Y>.

- [9] Thomas Debris-Alazard та Jean-Pierre Tillich. «Two attacks on rank metric code-based schemes: RankSign and an IBE scheme». B: *Advances in Cryptology—ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 3-7, 2018, Proceedings, Part I 24*. Springer. 2018, c. 87—116.
- [10] Amos Fiat та Adi Shamir. «How to prove yourself: Practical solutions to identification and signature problems». B: (1987), c. 186—194.
- [11] Philippe Gaborit та Marc Girault. «Lightweight Code-Based Identification and Signature». B: (2007), c. 191—195. DOI: 10.1109/ISIT.2007.4557225.
- [12] *IEEE Standard for Low-Rate Wireless Networks*. Standard. (Revision of IEEE Std 802.15.4-2011). Institute of Electrical та Electronics Engineers, 2016. DOI: 10.1109/IEEESTD.2016.7460875. URL: <https://iee.org>.
- [13] Jonathan Katz та Yehuda Lindell. *Introduction to Modern Cryptography: Revised Third Edition*. CRC Press, 2025. ISBN: 9781040366509. URL: <https://books.google.de/books?id=5TteEQAAQBAJ>.
- [14] National Institute of Standards and Technology. *Module-Lattice-Based Digital Signature Standard*. Tex. 3Bit. NIST, 2024. DOI: 10.6028 / NIST . FIPS . 204. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=958463.
- [15] National Institute of Standards and Technology. *Stateless Hash-Based Digital Signature Standard*. Tex. 3Bit. NIST, 2024. DOI: 10.6028 / NIST . FIPS . 205. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=958464.
- [16] Partha Sarathi Roy та ии. «Evaluation of Code-based Signature Schemes». B: (2019). URL: <https://eprint.iacr.org/2019/544>.

- [17] Yongcheng Song та ін. «Statistical zero-knowledge and analysis of rank-metric zero-knowledge proofs of knowledge». B: *Theoretical Computer Science* (2023). ISSN: 0304-3975. DOI: 10.1016/j.tcs.2023.113731. URL: <https://doi.org>.
- [18] Jacques Stern. «A New Identification Scheme Based on Syndrome Decoding». B: (1993), c. 13—21. DOI: 10.1007/3-540-48329-2_2.
- [19] Pascal Véron. «Improved Identification Schemes Based on Error-Correcting Codes». B: *Applicable Algebra in Engineering, Communication and Computing* (1996), c. 57—69.