**УДК 004.056.55** **DOI:10.30837/rt.2021.2.205.04**

*М.В. ЄСІНА, канд. техн. наук, Б.С. ШАХОВ*

# АНАЛІЗ АПАРАТНИХ РЕАЛІЗАЦІЙ АЛГОРИТМІВ ЕЛЕКТРОННОГО ПІДПИСУ QTESLA, CRYSTALS-DILITIUM І MQDSS НА РІЗНИХ РІВНЯХ БЕЗПЕКИ

## Вступ

Як відомо, існуючі алгоритми криптографії з відкритим ключем, що засновані на RSA та еліптичних кривих, надають гарантії безпеки, які супроводжуються складністю; також можна казати про неможливість вирішення завдань цілочисельної факторизації і дискретного логарифму. Але експерти прогнозують, що створення квантового комп'ютера зможе зламати класичні криптографічні алгоритми. Через цю майбутню проблему Національний інститут стандартів і технологій (NIST) разом із провідними вченими у галузі криптографії розпочав відкритий процес стандартизації алгоритмів з відкритим ключем для квантових атак. Метою цієї роботи є аналіз та порівняння трьох апаратних реалізацій кандидатів 2-го раунду конкурсу NIST на алгоритм електронного підпису.

Криптографія з відкритим ключем – це фундаментальний протокол безпеки для всіх форм цифрового зв'язку, дротового або бездротового. Криптографія з відкритим ключем має три основні криптографічні функції, а саме: шифрування з відкритим ключем, електронні підписи (ЕП) і обмін ключами. Алгоритми криптографії з відкритим ключем на основі RSA і еліптичних кривих забезпечують гарантії безпеки, засновані на складності вирішення завдань цілочисельної факторизації і дискретного логарифму. Пітер Шор показав, що квантові комп'ютери можуть факторизувати цілі числа за поліноміальний час, роблячи традиційні алгоритми криптографії з відкритим ключем неефективними. Після цього криптографи стали шукати надійні альтернативи, такі як криптографічні алгоритми на основі решіток та математичних кодів.

З історичної точки зору, у 1997 р. NIST запросив рекомендації у громадськості для визначення заміни стандарту шифрування даних (DES), Advanced Encryption Standard (AES) [4]. Відтоді відкриті криптографічні оцінки стали способом вибору криптографічних стандартів. Наприклад, NESSIE (2000-2002), eSTREAM (2004-2008), CRYPTREC (2000-2002), SHA-3 (2007-2012) і CAESAR (2013-2019) прийняли цей підхід. У цих оцінках головним параметром була безпека. Продуктивність у програмному забезпеченні, продуктивність у прикладних специфічних інтегральних схемах (ASIC), продуктивність у FPGA та можливість реалізації з використанням обмежених ресурсів (невеликих мікропроцесорів та малопотужних апаратних засобів) є вторинними критеріями. У конкурсі AES Rijndael мав найшвидшу ASIC і другу найшвидшу реалізацію FPGA з тими ж гарантіями безпеки, що й конкуренти [5].

У роботі описується порівняння апаратного забезпечення трьох алгоритмів підпису (qTesla, Crystals-Dilitium, MQDSS), які зокрема є кандидатами 2-го раунду конкурсу NIST PQC, а алгоритм Crystals-Dilitium – фіналістом цього конкурсу.

## 1. Постквантова криптографія

Експерти та вчені у галузі криптографії почали розробляти постквантові криптографічні алгоритми, які зможуть вистояти проти атак постквантових комп'ютерів. Безперечно, алгоритми постквантової криптографії поділяються на класи. У цій роботі розглядаються два алгоритми на основі алгебраїчних решіток (qTesla та Crystal-Dilitium) і один алгоритм на основі багатовимірної криптографії (MQDSS) [1 - 8].

Криптографія на основі решіток заснована на підході побудови алгоритмів асиметричного шифрування з використанням задач теорії решіток, тобто задач оптимізації на дискретних адитивних підгрупах, що задані у множині $\mathbb{R}^n$. Такі алгоритми забезпечують найкращу продуктивність, але є найменш консервативними серед усіх. Криптографія на основі решіток ґрунтується на вирішенні наступних обчислювально важких задач:

- Задача знаходження найкоротшого вектору (SVP, Shortest Vector Problem) – знайти в заданому базисі решітки ненульовий вектор по відношенню до визначеної нормалі.
- Задача знаходження ідеального найкоротшого вектору (ISVP, Ideal Shortest Vector Problem) не вважається NP-складною. Однак не існує відомих решіток, що засновані на методі редукції, значно більш ефективних на ідеальних структурах, чим на загальних.
- Задача знаходження (приблизно) найкоротшого незалежного вектору (SIVP, Shortest Independent Vector Problem), в якій є базис решітки $B$ і необхідно знайти $n$ лінійно незалежних векторів.
- Задача знаходження найближчого вектору (CVP, Closest Vector Problem) – знаходження вектору за заданим базисом і деяким вектором, який не належить решітці, при цьому максимально схожий за довжиною з заданим вектором.

Навіть з квантовим комп'ютером SVP представлений багаточленом у ступені $n$. Інші криптографічні алгоритми на основі алгебраїчних решіток засновані на проблемі короткого цілочисельного рішення (SIS).

Багатовимірна криптографія або багатовимірна криптографія відкритого ключа – це загальний термін, що описує асиметричні криптографічні схеми, побудовані на вирішенні рівнянь заснованих на багатовимірних поліномах над кінцевим полем $F$. Безпека багатовимірної криптографії заснована на припущенні, що вирішення системи квадратичних багаточленів над кінцевим полем $F$, в загальному випадку, є NP-повною задачею в сильному сенсі або просто NP-повна. Ці схеми часто вважаються гарними кандидатами для постквантової криптографії також через те, що вони пропонують найкоротші підписи.

Існують й інші класи постквантових криптографічних алгоритмів, такі як: криптографія на основі математичних кодів, ЕП на основі гешування та інші криптографічні методи. Але у даній роботі розглядаються лише алгоритми, що засновані на багатовимірній криптографії та криптографії на основі решіток, через те, що вони є найбільш використовуваними, що можна зрозуміти з табл. 1, де показано, що алгоритми ЕП цих двох класів мають найбільше представників у конкурсі NIST.

Таблиця 1

Математика ЕП і KEM, що представлені на конкурс PQC

| Математика | Проблема, що вирішується | Підписи | KEM | Всього |
|---|---|---|---|---|
| Решітки | Пошук найкоротшого вектору, найбільш близького вектору | 5 (3) | 23 (9) | 28 (12) |
| Коди | Декодування випадкового лінійного коду | 3 (0) | 17 (7) | 20 (7) |
| Багатовимірна поліноміальна | Розв'язання багатовимірних квадратичних рівнянь | 8 (4) | 2 (0) | 10 (4) |
| Геш | Стійкість до атаки знаходження прообразу геш-функції | 3 (2) | 0 (0) | 3 (2) |
| Ізогенії | Пошук ізогенного відображення між еліптичними кривими з однаковою кількістю точок | 0 (0) | 1 (1) | 1 (1) |
| Інші | – | 2 (0) | 5 (0) | 7 (0) |
| Всього | – | 21 (9) | 48 (17) | 69 (26) |

Наведена кількість алгоритмів для оцінки NIST PQC раунду 1 і раунду 2 (всередині фігурних дужок).

### 1.1. Електронні підписи

Поточний процес стандартизації PQC NIST консолідує невразливих кандидатів після кожного наступного раунду. Кожен кандидат у конкурсі PQC NIST реалізує одну з трьох функцій: шифрування відкритого ключа, електронний підпис і механізм інкапсуляції ключа (KEM). Табл. 1 показує кількість представлень PQC раунду 2 і підсумовує їх функціональність і математичну складність.

NIST розраховує стандартизувати кілька алгоритмів, щоб забезпечити різні компроміси залежно від програми (швидкість і потужність пам'яті тощо). Пряме порівняння гарантій безпеки, що забезпечуються алгоритмами PQC-кандидатів, є складним завданням у відсут-

ність стандартної платформи квантових обчислень, відмінностей у математичних функціях, що лежать в основі алгоритмів, і складного алгоритму специфікації порівняно з попередніми криптографічними конкурсами [2].

Електронні підписи PQC працюють над принципом, що відправник підписує повідомлення за допомогою закритого ключа, а одержувач перевіряє підпис, використовуючи відкритий ключ відправника. Ці алгоритми використовують три функції:

- *crypto_sign_keypair* генерує відкритий ключ *pk* та секретний ключ *sk*;
- *crypto_sign* приймає *sk* і повідомлення *m* плюс його довжина *mlen* і видає підпис *sm,* доданий до повідомлення;
- *crypto_sign_open* приймає *pk, sm* та довжину *smlen* і видає повідомлення *m.*

### 1.2. Класифікація безпеки алгоритмів PQC

NIST виділяє п’ять категорій потужності безпеки:
Перший рівень безпеки ⇒ еквівалентний пошуку ключа AES-128.
Другий рівень безпеки ⇒ еквівалентний пошуку колізій SHA-256/SHA3-256.
Третій рівень безпеки ⇒ еквівалентний пошуку ключа AES-192.
Четвертий рівень безпеки ⇒ еквівалентний пошуку колізій SHA-384/SHA3-384.
П’ятий рівень безпеки ⇒ еквівалентний пошуку ключа AES-256.
Рівні безпеки PQC алгоритмів другого раунду NIST наведені у табл. 2.

## 2. Оцінка апаратних засобів PQC

Алгоритми ЕП PQC, їх реалізація і характеристики безпеки узагальнені в табл. 2. Розглядається програмна реалізація із використанням Xilinx Virtex-7 FPGA.

Розглядаються такі показники продуктивності: затримка, область і область затримки. Затримка – це час, необхідний системі для отримання вихідного сигналу з моменту подачі вхідного сигналу. Пропускна здатність – це максимальна швидкість, з якою можна забезпечити виведення. Мінімальна кількість тактових циклів між двома послідовними входами є інтервалом ініціювання і є мірою пропускної здатності [2].

Таблиця 2

Алгоритми ЕП PQC, які досліджуються

| Алгоритм | Вирішувальна задача | Примітив PQC | Підтримуваний рівень безпеки (розмір відкритого ключа в байтах) | | | | |
|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 |
| Crystals-Dilithium [6] | Решітки | Алгоритм ЕП | **1184** | 1472 | 1760 | – | – |
| MQDSS [7] | Багатовимірна криптографія | Алгоритм ЕП | – | **62** | – | 88 | – |
| qTESLA [8] | Решітки | Алгоритм ЕП | **1504**<br>14880 | – | 3104<br>2976<br>39712 | – | – |

Алгоритми, що не підтримують визначений рівень безпеки, позначені символом "–".

У табл. 3 і 4 представлені службові дані апаратних засобів і синхронізації для реалізації алгоритмів підпису, верифікації і генерації ключової пари PQC, відповідно, при синтезі без будь-яких додаткових обмежень (затримки).

Таблиця 3

Безпека в порівнянні з площею залежно від часу алгоритмів підпису PQC,
без оптимізації (тобто базового рівня)

| Алгоритм | Рівень безпеки | Тригери | LUT | Тактовий цикл (нс) | Затримка |
|---|---|---|---|---|---|
| Алгоритми електронного підпису | | | | | |
| Crystals-Dilithium | 1 | 25926 | 133461 | 10 | 609828 |
| qTELSA | 1 | 41978 | 232582 | 10 | 125374 |
| MQDSS | 2 | 35263 | 193320 | 15 | 49365597 |

Серед алгоритмів підпису рівня безпеки 1 Crystals-Dilitium і qTesla має затримки менше мільйона циклів. Crystals-Dilitium (для генерації підпису) є хорошим кандидатом для пристроїв Інтернету речей. Він має рівень безпеки 1 і є другим найшвидшим і другим найменшим серед алгоритмів підпису.

Таблиця 4

Співвідношення безпеки і площі та часу алгоритмів верифікації PQC
без оптимізації (тобто базового рівня)

| Алгоритм | Рівень безпеки | Тригери | LUT | Тактовий цикл (нс) | Затримка |
|---|---|---|---|---|---|
| Алгоритми електронного підпису | | | | | |
| Crystals-Dilithium | 1 | 20865 | 108878 | 10 | 5380 |
| qTELSA | 1 | 29875 | 168570 | 10 | 71223 |
| MQDSS | 2 | 26423 | 147359 | 15 | 25124450 |

Серед алгоритмів підпису qTesla і Crystals-Dilitium мають затримку менше мільйона циклів. Crystals-Dilitium є найшвидшим серед алгоритмів підпису. Crystals-Dilithium (для створення підпису) є хорошими кандидатами для пристроїв Інтернету речей. Жодна з верифікацій безпеки рівня 5 в цьому дослідженні не має низької затримки і, отже, не підходить для серверів. Всі алгоритми з низькою затримкою мають безпеку рівня 1.

У табл. 5 наведено та розглянуто критичні функції і цикли, які призводять до високої затримки.

Таблиця 5

Критичні функції алгоритмів електронного підпису

| Алгоритм | Критична функція | # Цикли |
|---|---|---|
| Алгоритми електронного підпису | | |
| Crystals-Dilithium | *expand_mat* | 2 |
| MQDSS | *crypto_sign* | 2 |
| qTELSA | *sparse_mul16* | 1 |

Використовується розв'язка циклу для підпису і верифікації, результати наведені в табл. 6 і 7 відповідно. Crystals-Dilitium – найшвидший алгоритм підпису. Результати підпису з використанням конвеєрного циклу представлені в табл. 8. Серед алгоритмів з рівнем безпеки 1 Crystals-Dilitium – найшвидший алгоритм підпису.

Таблиця 6

Співвідношення безпеки і площі порівняно із синхронізацією алгоритмів підпису PQC,
після завершення циклу

| Алгоритм | Рівень безпеки | Тригери | LUT | Тактовий цикл (нс) | Затримка |
|---|---|---|---|---|---|
| Алгоритми електронного підпису | | | | | |
| Crystals-Dilithium | 1 | 158313 | 584742 | 10 | 18525 |
| qTELSA | 1 | 97235 | 328106 | 10 | 59854 |
| MQDSS | 2 | 45135 | 230273 | 15 | 49365597 |

Crystals-Dilitium посідає найбільше місце серед алгоритмів підпису. Серед алгоритмів підпису PQC розгортання циклу не зменшує затримки MQDSS. Розгортання циклу зменшує затримку всіх алгоритмів підпису PQC. Однак це також призводить до збільшення площі. Серед інших алгоритмів високої безпеки MQDSS (підпис, рівень безпеки 2) може використовуватися для пристроїв Інтернету речей, оскільки забезпечує відносно низьке апаратне навантаження.

Таблиця 7

Залежність безпеки від площі залежно від часу алгоритмів верифікації PQC,
після завершення циклу

| Алгоритм | Рівень безпеки | Тригери | LUT | Тактовий цикл (нс) | Затримка |
|---|---|---|---|---|---|
| Алгоритми електронного підпису | | | | | |
| Crystals-Dilithium | 1 | 108154 | 388991 | 10 | 5380 |
| qTELSA | 1 | 77567 | 247726 | 10 | 36423 |
| MQDSS | 2 | 93945 | 323734 | 15 | 25084906 |

Розгортання циклу забезпечує суттєве скорочення затримок. За винятком MQDSS, алгоритм не має затримки понад 1 мільйон циклів. Подібно підпису, розгортання циклу зменшує затримку верифікації PQC. Це поставляється з додатковим обладнанням. Для пристроїв Інтернету речей, де високий рівень безпеки не потрібен, можна використовувати алгоритм рівня безпеки 1 з низьким рівнем службових даних, наприклад, qTesla (підпис). З іншого боку, Crystals-Dilitium (підпис), що забезпечує безпеку рівня 1, може використовуватися в серверах через його низьку затримку.

Результати підпису з використанням конвеєрного циклу представлені в табл. 8.

Таблиця 8

Співвідношення безпеки і площі порівняно із синхронізацією алгоритмів ЕП PQC, після конвеєризації циклу

| Алгоритм | Рівень безпеки | Тригери | LUT | Тактовий цикл (нс) | Затримка |
|---|---|---|---|---|---|
| Алгоритми електронного підпису | | | | | |
| Crystals-Dilithium | 1 | 146076 | 1327355 | 10 | 155166 |
| qTELSA | 1 | 112657 | 346020 | 10 | 63736 |
| MQDSS | 2 | 47441 | 270713 | 15 | 25825918 |

Подібно до розгортання циклу, конвеєризація також зменшує загальну затримку для алгоритмів підпису PQC. Основна відмінність з таблицею 6 пов'язана з алгоритмом підпису MQDSS. Хоча при розгортанні циклу не вдалося змінити його затримку, конвеєрна обробка може скоротити затримку на 50 %. Конвеєрний цикл зменшує затримку алгоритмів підпису PQC, збільшуючи апаратну область. Поліпшення затримки порівняно з розгортанням циклу не погоджено. Жоден з алгоритмів підпису не забезпечує низьку затримку після конвеєрної обробки [2].

### 3. Поглиблені дослідження з використанням трьох підписів PQC

У цьому розділі розглянуто три алгоритми PQC на основі підпису і проаналізовано апаратні реалізації на різних рівнях безпеки. Крім того, розглянуто вплив різних оптимізацій конструкції як для процедури підпису, так і для процедури верифікації. Всі досліджувані реалізації виконані на платі Xilinx Artix-7 FPGA. Це платформа, яку NIST розглядає в якості платформи порівняння [2, 3].

#### 3.1. Порівняння за рівнем безпеки 1

Порівнюється область і продуктивність трьох алгоритмів для рівня безпеки 1. Аналізовані параметри – кількість тригерів, кількість LUT, затримка і LAP. Порівняння частин підпису та верифікації наведено в табл. 9 і 10 [2 – 4].

Таблиця 9

Аналіз апаратних реалізацій компонентів "crypto_sign" алгоритмів PQC на основі підпису

| Алгоритм | FF | LUT | Такти (нс) | Затримка | LAP |
|---|---|---|---|---|---|
| qTesla | 26299 | 126732 | 12.65 | 537092 | $6.8 \times 10^{10}$ |
| Crystals-Dilithium | 27132 | 123655 | 8.738 | 485963 | $6.0 \times 10^{10}$ |
| MQDSS | 21841 | 106035 | 17.05 | 34502428 | $3.6 \times 10^{10}$ |

У той час як накладні витрати на розмірі більш-менш однакові (MQDSS займає на 83 % менше місця, ніж інші), загальна затримка і LAP для MQDSS надзвичайно високі. MQDSS займає в 71 раз більше затримки і в 61 раз більше LAP порівняно з Crystals-Dilitium. На рівні безпеки 1, Crystals-Dilitium є найкращим алгоритмом для "crypto_sign", оскільки займає найменший LAP.

Таблиця 10

Аналіз апаратних реалізацій "crypto_sign_open" компонентів алгоритмів PQC на основі підпису

| Алгоритм | FF | LUT | Такти (нс) | Затримка | LAP |
|---|---|---|---|---|---|
| qTesla | 17780 | 87067 | 12.58 | 80422 | $7.0 \times 10^{9}$ |
| Crystals-Dilithium | 14712 | 63863 | 8.738 | 149950 | $9.5 \times 10^{9}$ |
| MQDSS | 23072 | 117097 | 17.045 | 25686731 | $3.0 \times 10^{12}$ |

Crystals-Dilitium має найменший розмір, тоді як qTesla має найменшу затримку і показує кращу продуктивність. На рівні безпеки 1 qTesla є найкращим алгоритмом для "crypto_sign_open", оскільки займає найменший LAP [2].

### 3.2. Порівняння за рівнем безпеки 3

У цьому розділі порівнюється вартість і продуктивність впровадження апаратного забезпечення як для "crypto_sign", так і для "crypto_sign_open" для трьох алгоритмів на основі підпису на рівні безпеки 3. Результати для підпису та верифікації наведено у табл. 11 та 12 відповідно.

Таблиця 11

Аналіз апаратних реалізацій "crypto_sign" компонентів алгоритмів PQC на основі підпису на рівні безпеки 3

| Алгоритм | FF | LUT | Такти (нс) | Затримка | LAP |
|---|---|---|---|---|---|
| qTesla | 26011 | 126311 | 12.65 | 347655 | $4.3 \times 10^{10}$ |
| Crystals-Dilithium | 27308 | 123933 | 8.738 | 826832 | $1.0 \times 10^{11}$ |
| MQDSS | 24748 | 123170 | 16.378 | 119353597 | $1.4 \times 10^{13}$ |

У той час як накладні витрати на розмір більш-менш однакові для всіх алгоритмів, загальна затримка і LAP для MQDSS надзвичайно високі. На рівні безпеки 3 qTesla є найкращим алгоритмом для "crypto_sign", оскільки займає найменший LAP [2, 3].

Таблиця 12

Аналіз апаратних реалізацій "crypto_sign_open" компонентів алгоритмів PQC на основі підпису

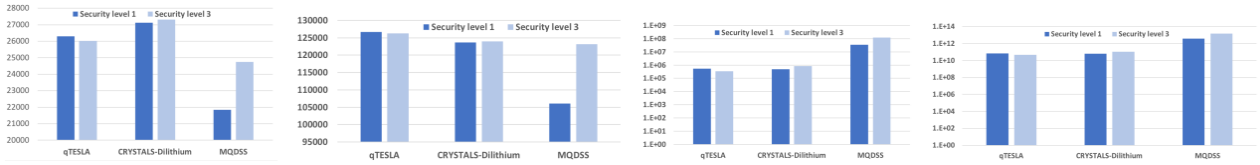
| Алгоритм | FF | LUT | Такти (нс) | Затримка | LAP |
|---|---|---|---|---|---|
| qTesla | 17754 | 86142 | 12.65 | 201027 | $1.7 \times 10^{10}$ |
| Crystals-Dilithium | 14783 | 63980 | 8.738 | 297592 | $1.9 \times 10^{10}$ |
| MQDSS | 23149 | 117574 | 17.045 | 87861777 | $1.0 \times 10^{13}$ |

Crystals-Dilitium має найменший розмір, тоді як qTesla має найменшу затримку. На рівні безпеки 3 qTesla є найкращим алгоритмом для "crypto_sign_open", оскільки займає найменший LAP. З іншого боку, MQDSS має найгірші показники LAP [2].

### 3.3. Співставлення між різними рівнями безпеки

У цьому розділі порівнюються службові дані області та продуктивність алгоритмів на різних рівнях безпеки [2].

Порівнюється область (у термінах тригерів і LUT), продуктивність (у термінах затримки) і LAP для реалізації "crypto_sign" частини трьох алгоритмів для рівнів безпеки 1 і 3. Результати показано на рис. 1. Для qTesla і Crystals-Dilitium площа накладних витрат на рівнях безпеки 1 і 3 аналогічна. Для MQDSS, накладні витрати на рівень безпеки 3 на 13 – 16 % більше, ніж на рівень безпеки 1. Для qTesla затримка знижується на рівні безпеки 3 порівняно з рівнем 1. Для двох інших алгоритмів затримка різко збільшується при більш високому рівні безпеки. Порівнюючи як площу, так і продуктивність, можна помітити, що тільки qTesla має більш низькі LAP на рівні безпеки 3 порівняно з рівнем 1. qTesla на рівні безпеки 3 має найнижчий LAP і забезпечує найвищу безпеку серед усіх альтернатив, що обговорюються в цьому підрозділі.




(a) (b) (c) (d)

Рис. 1. Порівняння (*a*) тригерів, (*b*) LUT, (*c*) затримки і (*d*) LAP для реалізації компонента "crypto_sign" трьох алгоритмів PQC на різних рівнях безпеки

Для qTesla і Crystals-Dilitium різниця в кількості тригерів не є істотною, в той час як для MQDSS кількість тригерів для реалізації на рівні безпеки 3 збільшується на 13% порівняно з рівнем безпеки 1. Для qTesla і Crystals-Dilitium різниця в кількості LUT не є істотною, в той час як для MQDSS кількість тригерів для реалізації на рівні безпеки 3 збільшується на 16 % порівняно з рівнем безпеки 1. Оскільки затримка MQDSS на два порядку більша, ніж qTesla або Crystals-Dilitium, цей графік зображено в логарифмічній шкалі. qTesla на рівні безпеки 3 і Crystals-Dilitium на рівні безпеки 1 мають найменшу затримку. qTesla на рівні безпеки 3 має найкращий LAP, а MQDSS на обох рівнях безпеки має найгірший LAP.

Далі порівнюється область, продуктивність і LAP для верифікації підпису, тобто компонент "crypto_sign_open" трьох алгоритмів на двох рівнях безпеки. Результати показано на рис. 2. Для всіх алгоритмів службові дані області залишаються однаковими для двох рівнів безпеки. З іншого боку, послідовно затримка і LAP вища на рівні безпеки 3 порівняно з рівнем безпеки 1. qTesla має найменший LAP на обох рівнях безпеки.

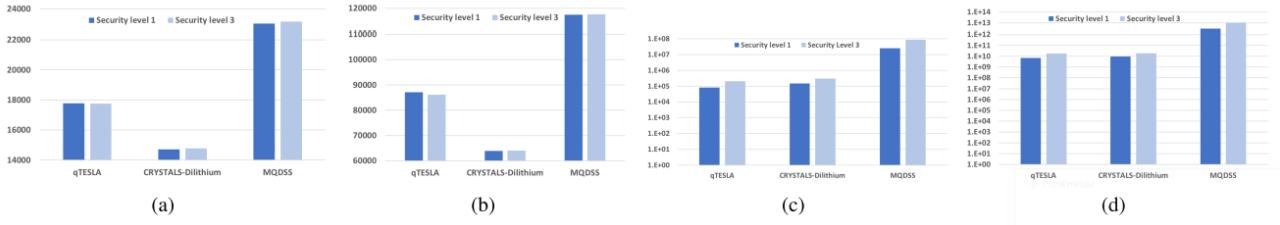



Рис. 2. Порівняння (*a*) FF, (*b*) LUT, (*c*) затримки і (*d*) LAP для реалізації компонента "crypto_sign_open" трьох алгоритмів PQC на різних рівнях безпеки

У той час як на рис. 1 Crystals-Dilitium займає найбільшу кількість тригерів, при реалізації "crypto_sign_open", потрібна найменша кількість тригерів. Порівняно з рис. 1 різниця в кількості LUT не є значною для жодного алгоритму. qTesla на рівні безпеки 1 вимагає найменшої затримки. MQDSS вимагає найбільших затримок для "crypto_sign" і "crypto_sign_open." qTesla на рівні безпеки 1 має найкращий LAP, а MQDSS на обох рівнях безпеки має найгірший LAP [2].

### 3.4. Оптимізація

У цьому підрозділі проаналізовано, як різні методи оптимізації (розбиття циклу і конвеєрування циклу) допомагають зменшити загальну затримку алгоритмів PQC. Результати реалізації "crypto_sign" і "crypto_sign_open" показані на рис. 3 і 4 відповідно.

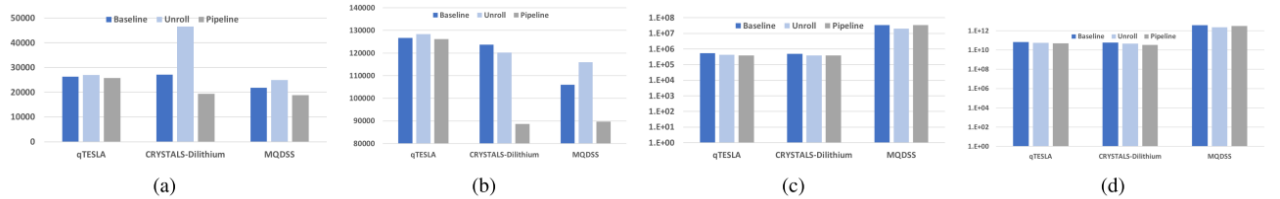



Рис. 3. Порівняння (*a*) FF, (*b*) LUT, (*c*) затримки і (*d*) LAP для реалізації "crypto_sign" трьох алгоритмів PQC на рівні безпеки 1 для різних оптимізацій

Конвеєрний цикл дає найменшу кількість тригерів по всьому циклу, в той час як розгін циклу вимагає максимальної кількості тригерів. Конвеєрний цикл дає найменшу кількість LUT для всіх конструкцій. За винятком Crystals-Dilitium, кількість LUT збільшується з розмотуванням циклів. Розгортання циклу і конвеєрна обробка зменшують затримку порівняно з базовою лінією. Crystals-Dilitium і qTesla мають найменше значення LAP при конвертуванні циклів. Для MQDSS найменше значення LAP виходить, коли цикли не розгорнуті [2, 3].

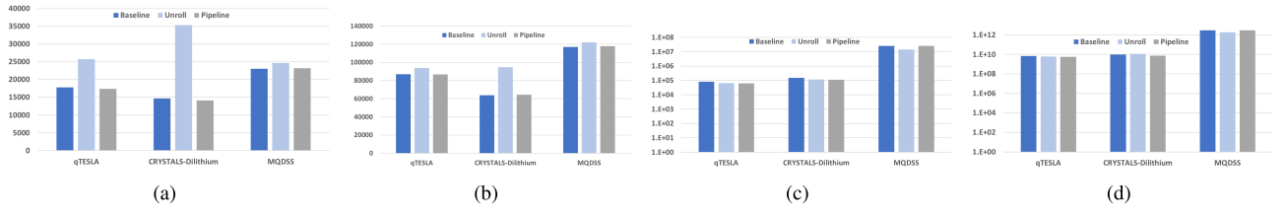



Рис. 4. Порівняння (*a*) FF, (*b*) LUT, (*c*) затримки і (*d*) LAP для реалізації компонента "crypto_sign_open" трьох алгоритмів PQC на рівні безпеки 1 для різних оптимізацій

Оптимізація з конвеєрними циклами займає найменшу кількість тригерів і LUT у всіх конструкціях, в той час як розукрупнення циклу вимагає максимальної кількості тригерів і LUT. Оптимізація за допомогою розмотування циклів і конвеєрів дозволяє скоротити затримки порівняно з базовою реалізацією. За винятком MQDSS, два інших алгоритми мають найменше значення LAP при виконанні конвеєризації циклу. Для MQDSS найменше значення LAP виходить, коли цикли не розгорнуті [2].

### 3.5. Наслідки апаратного оптимізованого Keccak

Keccak – сімейство sponge-функцій, що використовуються три алгоритми другого раунду NIST PQC. Протягом багатьох років дослідники розробляли апаратно оптимізовані варіанти Keccak. У цьому підрозділі розглянемо вплив одного з них і спостерігаємо, як змінюється розмір і накладні витрати на продуктивність. Використовується реалізація Keccak [3]. Автори [3] розвивали впровадження функції *KeccakF1600_StatePermute* з окремими функціями, що використовуються для внутрішніх операцій як $\theta$, $\rho$, $\pi$, $\chi$ та $\iota$. На рис. 5 порівнюються параметри для реалізації "crypto_sign", а на рис. 6 порівнюється "crypto_sign_open". Для забезпечення узгодженості всі реалізації належать до рівня безпеки 1. На рис. 7, 8 порівнюються накладні витрати з розмотуванням циклів. Для чесного порівняння використовуються одні й ті ж директиви для обох реалізацій. Аналогічне порівняння для конвеєризації циклу показано на рис. 9, 10 [3].

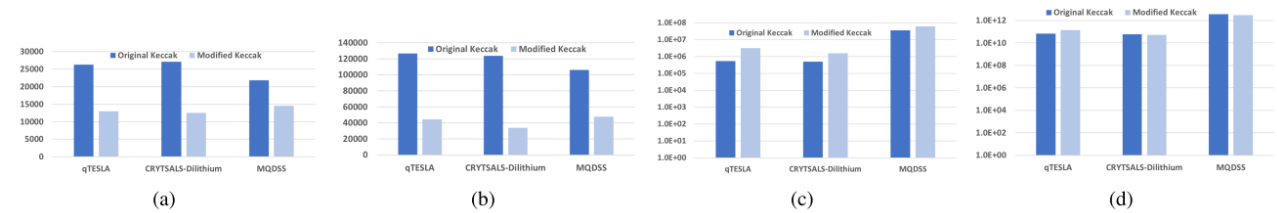



Рис. 5. Порівняння (*a*) FF, (*b*) LUT, (*c*) затримки і (*d*) LAP для реалізації компонента "crypto_sign" трьох алгоритмів PQC на рівні безпеки 1 для двох версій Keccak

Кількість тригерів і LUT значно зменшується порівняно з початковим Keccak. Crystals-Dilitium має 54 % зниження кількості тригерів і 73% зниження LUT. З іншого боку, затримка збільшується з модифікованим Keccak. Найсильніше постраждав qTesla, який має збільшення затримки в 5,8 разів, в той час як MQDSS несе тільки збільшення затримки в 1,7 рази. З іншого боку, значення LAP зменшується для MQDSS і Crystals-Dilitium і збільшується тільки для qTesla. Для MQDSS LAP скорочується на 21 % [3].

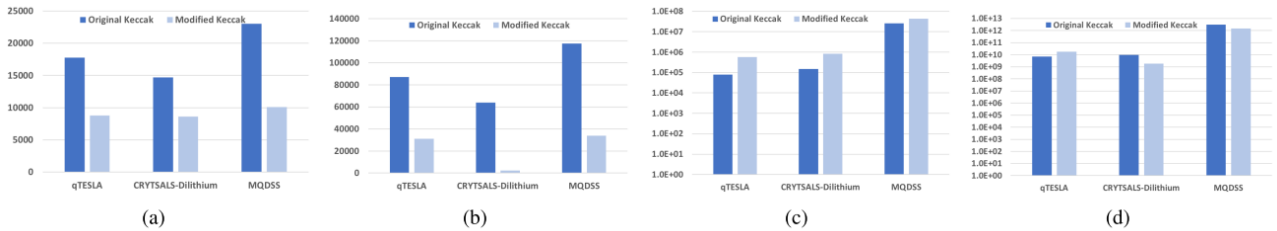



Рис. 6. Порівняння (*a*) тригерів, (*b*) LUT, (*c*) затримки і (*d*) LAP для реалізації компонента "crypto_sign_open" трьох алгоритмів на рівні безпеки 1 для різних версій Keccak

Як і на рис. 5, кількість тригерів і LUT значно зменшується порівняно з вихідним Keccak. MQDSS має максимальне 56 % зниження кількості тригерів і Crystals-Dilitium має близько 96 % зниження LUT. Затримка збільшується з модифікованим Keccak. Найсильніше це стосується qTesla, яка має збільшення затримки в 7 разів, в той час як MQDSS має збільшення затримки в 1,7 рази. LAP значно зменшується для всіх алгоритмів, крім qTesla. Crystals-Dilitium має 82 % зниження LAP порівняно з вихідним Keccak.

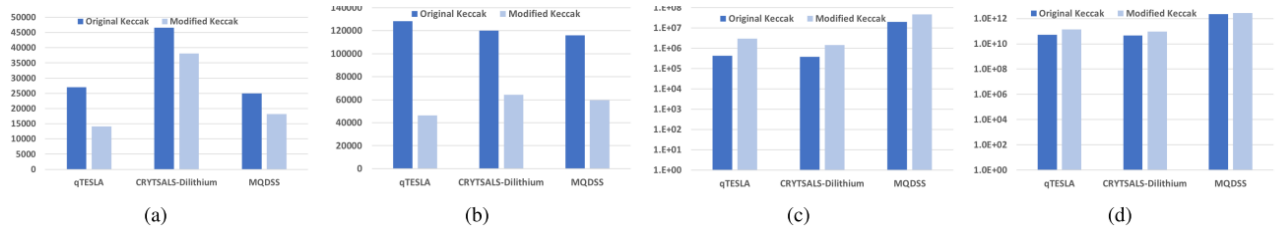



Рис. 7. Порівняння (*a*) тригери, (*b*) LUT, (*c*) Latency і (*d*) LAP для реалізації компонента "crypto_sign" трьох алгоритмів PQC на рівні безпеки 1 для різних версій Keccak після завершення циклу

Кількість тригерів і LUT значно зменшується порівняно з оригінальною версією Keccak. qTesla має приблизно 48 % зниження кількості тригерів і 64% скорочення кількості LUT. З іншого боку, затримка збільшується з реалізацією модифікованого Keccak. Найсильніше вражає qTesla, яка має збільшення затримки в 7,15 разів. На відміну від рис. 5, значення LAP послідовно збільшуються для всіх трьох алгоритмів. Для MQDSS збільшення LAP мінімально – 20 % [3].

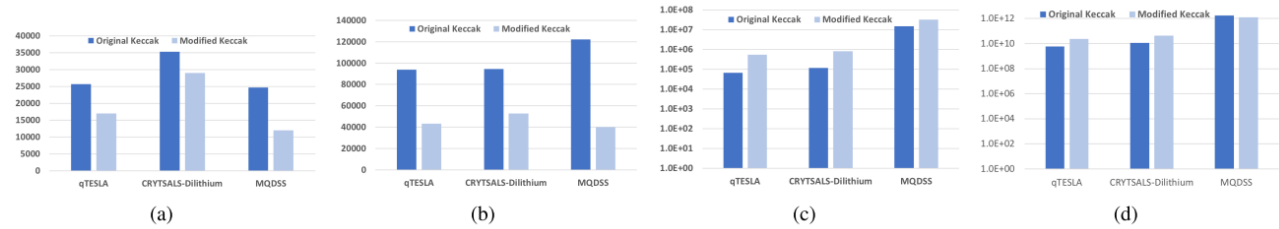



Рис. 8. Порівняння (*a*) тригери, (*b*) LUT, (*c*) LAP і (*d*) LAP для реалізації компонента "crypto_sign_open" трьох алгоритмів PQC на рівні безпеки 1 для різних Keccak після розгортання циклу

Кількість тригерів і LUT значно зменшується порівняно з початковим Keccak. MQDSS має приблизно 52 % зниження кількості тригерів і 61 % зменшення кількості LUT. З іншого боку, затримка збільшується з модифікованим Keccak. Найсильніше постраждав qTesla, який має збільшення в 8,4 рази. LAP збільшується для qTesla і Crystals-Dilitium, але знижується на 28 % для MQDSS.

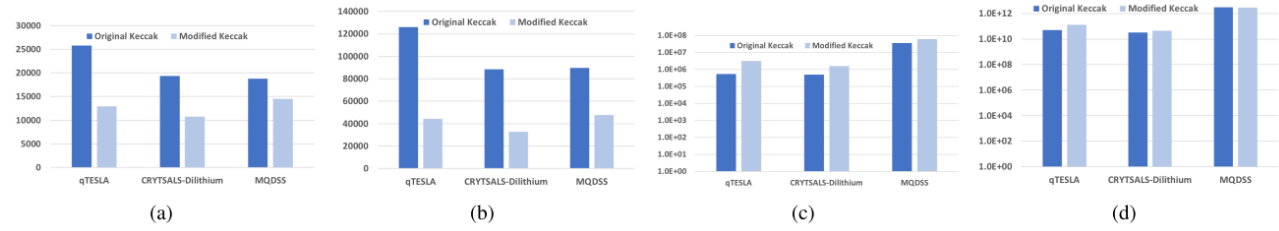



Рис. 9. Порівняння (*a*) тригери, (*b*) LUT, (*c*) затримка і (*d*) LAP для реалізації компонента "crypto_sign" трьох алгоритмів PQC на рівні безпеки 1 для різних версій Keccak, після конвеєризації циклу

Скорочення кількості тригерів і LUT значно порівняно з оригінальним Keccak. qTesla має 50 % зниження кількості тригерів і 65 % скорочення кількості LUT. З іншого боку, затримка збільшується з реалізацією модифікованого Keccak. Найсильніше це торкнулося

qTesla, що має збільшення затримки в 7,7 разів. Значення LAP збільшуються для qTesla і Crystals-Dilitium, але зменшуються на 5 % для MQDSS. В цілому, відносне збільшення LAP менше, ніж при розгортанні циклу на рис. 7.

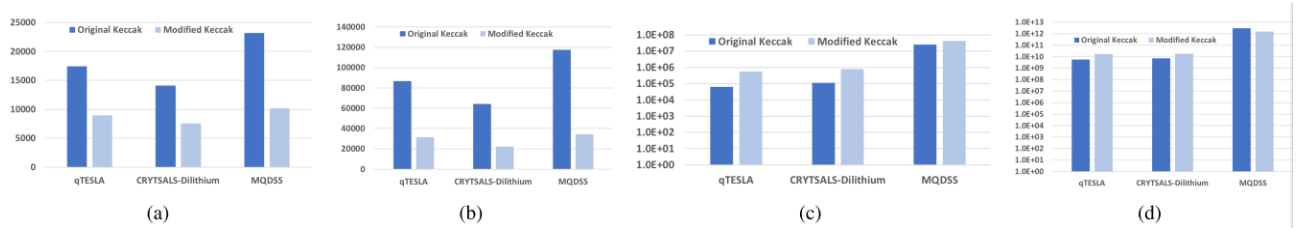



Рис. 10. Порівняння (*a*) тригери, (*b*) LUT, (*c*) затримка і (*d*) LAP для реалізації "crypto_sign_open" з трьох алгоритмів на рівні безпеки 1 з використанням різних версій Keccak, після конвеєризації циклу

Кількість тригерів і LUT значно зменшується порівняно з базовим Keccak. MQDSS має 56 % зниження кількості тригерів і 70 % скорочення кількості LUT. Хоча затримка MQDSS збільшується на 67 % при модифікованому Keccak, загальне значення LAP зменшується на 52 % [2].

**Висновки**

1. Наразі однією із важливих проблем сучасної криптографії є створення стандартів асиметричних криптоперетворень ЕП, які були б безпечними у постквантовий період. Вирішення цієї проблеми здійснюється в процесі міжнародного конкурсу NIST, завданням якого є розробка такого ЕП, який би був стійким як до квантових, так і до класичних атак.
2. Були проведені та проаналізовані специфічні тематичні дослідження з використанням трьох алгоритмів підпису – qTesla, Crystals-Dilitium і MQDSS. Відзначено розмір і продуктивність накладних витрат, коли апаратні реалізації цих алгоритмів виконуються для різних рівнів безпеки, а також для різних оптимізацій.
3. Алгоритми ЕП qTesla і Crystals-Dilitium мали найменший LAP при конвеєрному циклі, в той час як MQDSS мав найменший LAP при розгортанні циклу. Майбутні дослідження визначать, яка оптимізація підходить для якого алгоритму.
4. Навіть, якщо одні й ті самі директиви оптимізації застосовуються до двох різних версій Keccak, модифікована функція Keccak зменшує загальні витрати області. Однак у цьому випадку LAP знижуються не так сильно (фактично LAP збільшується в більшості випадків), порівняно з базовою лінією (без оптимізації). Зменшення LAP більше для конвеєрного циклу порівняно з розмотуванням циклу.
5. У цьому дослідженні було використано одну і ту ж плату (Virtex або Artix) для реалізації варіантів алгоритмів PQC як з маленьким розміром, так і з низькою затримкою. У майбутньому буде вивчено, які архітектури FPGA ідеально підходять для реалізації з маленьким розміром/низькою затримкою.

**Список літератури:**

1. PQC Standardization Process: Third Round Candidate Announcement. July 22, 2020. [Electronic resource]. Access mode: https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement.
2. Kanad Basu NIST Post-Quantum Cryptography – A Hardware Evaluation Study / Kanad Basu, Deepraj Soni, Mohammed Nabeel, Ramesh Karri // Access mode: https://eprint.iacr.org/2019/047.pdf.
3. K. Gaj Comprehensive comparison of hardware performance of fourteen round 2 SHA-3 candidates with 512-bit outputs using field programmable gate arrays / K. Gaj, E. Homsirikamol, M. Rogawski // 2nd SHA-3 Candidate Conference, Santa Barbara, August, pp. 23–24, 2010.
4. J. Nechvatal Report on the development of the advanced encryption standard (AES) / J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, E. Roback // 2001. Access mode: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4863838/.
5. K. Aoki Fast implementations of AES candidates. / K. Aoki, H. Lipmaa // AES Candidate Conference, pp. 106–120, 2000.
6. L. Ducas Crystals-Dilithium: Digital signatures from module lattices / L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, D. Stehle // Access mode: https://pq-crystals.org/dilithium/index.shtml, 2018.

7. Ming-Shing Chen From 5-pass MQ-based identification to MQ-based signatures / Ming-Shing Chen, A. Hulsing, J. Rijneveld, S. Samardjiska, P. Schwabe // Access mode: https://eprint.iacr.org/2016/708.pdf.

8. Erdem Alkim The Lattice-Based Digital Signature Scheme qTesla / Erdem Alkim, Paulo S. L. M. Barreto, Nina Bindel, Juliane Kramer, Patrick Longa, Jefferson E. Ricardini // Access mode: https://eprint.iacr.org/2019/085.pdf.

*Надійшла до редколегії 07.04.2021*

*Відомості про авторів:*

**Єсіна Марина Віталіївна** – канд. техн. наук, Харківський національний університет імені В.Н. Каразіна, старший викладач кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, Україна; e-mail: rinayes20@gmail.com; ORCID: https://orcid.org/0000-0002-1252-7606

**Шахов Богдан Сергійович** – Харківський національний університет імені В.Н. Каразіна, студент кафедри безпеки інформаційних систем і технологій, факультету комп'ютерних наук, Україна; e-mail: bogdanshahov2000@gmail.com