УДК 004.056.55 **DOI:10.30837/rt.2020.3.202.04**

Ю.І. ГОРБЕНКО, канд. техн. наук, О.С. ДРОЗДОВА

АНАЛІЗ СТІЙКОСТІ ПОСТКВАНТОВОГО ЕЛЕКТРОННОГО ПІДПИСУ DILITHIUM ДО АТАК НА ПОМИЛКИ

Вступ

На сьогодні поряд із задачею стандартизації постквантових алгоритмів стоїть задача поступового заміщення класичних алгоритмів, які використовуються у реальних додатках та пристроях на квантовостійкі. Критичною задачею у цьому процесі переходу є забезпечення відповідного рівня доказової стійкості та захищеності від атак не тільки в теорії, але і від атак на реалізацію та атак по стороннім каналам. Ця стаття присвячена дослідженню саме такої, практичної, стійкості Dilithium до атак на помилки.

Схема ЕП Dilithium [1] пройшла до третього раунду процесу постквантової стандартизації NIST в якості одного з трьох фіналістів. Статус «фіналіст» означає, що дана схема придатна до застосування у багатьох додатках,та випадках використання [2]. Характерними рисами Dilithium є маленькі розміри ключів та підпису, достатня швидкість, простота та гнучкість(адаптація до різних рівнів стійкості).

Очікується що незабаром після закінчення третього раунду відбудеться процес стандартизації одного з двох фіналістів ЕП (Dilithium або Falcon, обидві схеми засновані на математиці алгебраїчних решіток).

1. Основні відомості про стійкість Dilithium

Доцільно сказати, що теоретична та практична стійкість повинні доповнювати одна іншу, тому спочатку наведемо рівні доказової стійкості для України та сумісність з ними Dilithium.

Для використання в Україні пропонуються наступні чотири рівні доказової стійкості (класична / квантова):

0-й рівень – 128 біт / 64 біт

1-й рівень – 256 біт / 128 біт

2-й рівень – 384 біт / 192 біт

3-й рівень – 512 біт / 256 біт

Dilithium забезпечує 1й(AES128), 2й(SHA256/ SHA3-256) та 3й(AES192) рівні стійкості згідно NIST. Наприклад стійкість AES128 означає, що зламати схему так само важко як зламати 128-бітний AES повним перебором.

Важливо розуміти можливості таких атак на помилки на алгоритми ЕП, але до того як буде прийнято рішення щодо стандартизації постквантового ЕП в Україні, щоб забезпечити їх безпечне впровадження у перехідний та постквантовий періоди.

Стійкість Dilithium заснована на складності задачі Модульного навчання з помилками (MLWE) та Модульного короткого цілого рішення (MSIS). Схема підпису Dilithium використовує структуру Фіат – Шаміра з перериваннями. Структура Фіат – Шаміра потребує використання нонсу (одноразового випадкового значення), через це вона має добре відому вразливість – при використанні однакового нонсу для різних повідомлень, стає можливим відновити особистий ключ. Автори Dilithium запропонували контр захід до цієї атаки – вироблення нонсу шляхом гешування повідомлення та ключа, тож кожен раз нонс є унікальним. Таке рішення забезпечує детермінованість схеми ЕП, однак воно не стає у нагоді, коли зловмисник застосовує атаку на помилки. Він може ввести обчислювальну помилку в одному з підписів, тож два повідомлення будуть мати однаковий нонс.

2. Параметри Dilithium

Спершу наведемо основні параметри Dilithium, які знадобляться при описі атаки на помилки, причому c, h, z – частини підпису, де:

- c – значення після застосування гешування;
- σ – цифровий підпис;
- s – малий елемент особистого ключа;
- y – шум(компонент маскування;
- tr – геш-значення відкритого ключ.

Функція XOF у підписі Dilithium застосовується для детерміновано розширення початкового значення матриці A, використовуючи функцію розширюваного виводу SHAKE128. Ця функція зменшує розміри відкритого та особистого ключів, оскільки потрібно зберігати тільки ρ замість повного значення матриці A .

3. Сутність атаки на помилки

Атаки на помилки вперше були представлені в 1996 р. Боне та ін. [3] для реалізації атаки на RSA-CRT, а незабаром Біхам і Шамір описали диференціальну атаку на помилки стосовно шифру DESblock [4]. Атаки на помилки – це розповсюджений метод криптоаналізу, який полягає у внесенні помилок на будь-якому етапі дії електронної схеми, у нашому випадку, схеми ЕП Dilithium. Даний тип атак належить до атак по стороннім каналам.

Атаки на помилки на електронні системи вивчалися більше 40 років. Відтоді до реального обладнання застосовувались різні форми внесення помилок, такі як зміна напруги, подача високих температур на апаратні засоби, використання рентгенівських променів тощо [5]. Через необхідність фізичного доступу до пристрою внесення помилок в основному загрожували захищеному відмовостійкому апаратному забезпеченню, як смарт-картки, так і жорстким апаратним модулям безпеки. Це змінилося з появою апаратної помилки Rowhammer, що впливає на основну пам'ять комп'ютера [6]. У 2014 р. Кім та інші показали, що зловмисне програмне забезпечення може використовувати пошкодження пам'яті, щоб викликати обернення бітів у сусідніх областях пам'яті, минаючи захист пам'яті обладнання [6]. За допомогою Rowhammer помилки стало можливо вносити дистанційно і за допомогою лише програмного забезпечення. Такий спосіб став потужною можливістю внесення помилки.

Сутність диференційної атаки на помилки на схему Dilithium заключається у наступному. Зловмисник створює таку ситуацію, в якій жертва двічі підписує одне і те ж повідомлення, але в одному з підписів внесена обчислювальна помилка (наприклад, за рахунок збоїв годинника). Це призводить до того, що різні підписи використовують однаковий нонс, таким чином стає можливим відновити особистий ключ. При диференційних атаках на помилки різниця між помилковим та правильним виводом використовується для визначення інформації про особистий ключ.

4. Можливості здійснення атаки на помилки

Спочатку визначимо якою може бути помилка. Помилкою може бути пропуск інструкцій, арифметична помилка, збій у сховищі тощо. Помилки можуть застосовуватися протягом великого періоду часу виконання, а не лише у конкретних операціях.

Коротко проведення атаки можна описати наступним чином. Підписувач двічі виконує підписання одного повідомлення M. Перший раз помилка не вноситься, та отримуємо дійсний і коректний підпис $\sigma = (\mathrm{z}, \mathrm{h}, c)$. Другий раз вноситься помилка (змінні при підписі з помилкою позначаються рискою $'$). Більш точно, помилка вноситься така, що за рахунок детермінованості y' є непорушеним та дорівнює y, проте $c' \neq c$, таким чином $\mathrm{z}' = \mathrm{y} + c'\mathrm{s}_1$.

Однак структура Фіат – Шаміра з перериванням створює додаткові перешкоди для здійснення атаки. Існує вимога, щоб обчислення як дійсного, так і помилкового підпису закінчу-

валось однаковою ітерацією циклу переривання. Іншими словами, коли k_f позначає кінцеве значення лічильника циклу k, потрібно, щоб $\Delta k_f = k_f - k'_f = 0$. Зауважимо, що в алгоритмі підпису лічильник циклу k вводиться до детермінованої вибірки. Отже, щоб виконувалась тотожність $y = y'$ постає вимога $\Delta k_f = 0$. Через проміжні помилки та вплив тестів відхилення це, очевидно, не гарантується.

Далі описуються можливі сценарії здійснення атаки на помилки.

4.1. Випадкова помилка при зверненні до геш-функції

При даному способі внесення помилки до обчислення $c \in B_{60} := \mathrm{H}(\mu \| \mathrm{w}_1)$ вводиться випадкова помилка. Наприклад, можна використати один із входів μ, w_1 безпосередньо перед їх використанням у H, або внести помилку безпосередньо у саму геш-функцію H.

У [7] показано, що зловмисник може ввести помилку в правильну ітерацію k_f, тобто в останню при обчисленні помилки. Якщо крок відхилення проходить з різним c', секретний елемент s_1 може бути відновлений.

Атака відбулась успішно, якщо відновлений елемент особистого ключа s_1 є малим, а Δc є оборотним. Це вірно з дуже високою ймовірністю. Частка оборотних поліномів у R_q дорівнює $(1-1/q)^n$ [8], що для параметрів Dilithium приблизно становить $1-2^{-15}$. Знаючи s за допомогою лінійної алгебри можна відновити повний секретний ключ. Також можна використати малу норму $c\mathrm{s}$ обчислюючи $\|\Delta \mathrm{z}\|_2$ (або $\|\Delta \mathrm{z}\|_2$) і перевірити, чи вона не нижче певного порогу.

4.2. Випадкова помилка у множенні поліномів геш-функції

Ця помилка зводиться до зміни складової підпису $c := \mathrm{H}(\mu, \mathrm{w}_1)$ у геш-значенні H, яке використовується при обчислення її вхідних параметрів μ, w_1. Геш-значення повідомлення або відкритого ключа μ також використовується як початкове значення (seed) для детермінованої вибірки, отже, помилки в обчисленні $\mu := \mathrm{CRH}(tr \| M)$ не можна використати. Але можна використати помилки в обчисленні $\mathrm{w} := \mathrm{Ay}$, які призводять до неправильного w_1. Необхідні множення поліномів у q можна ефективно реалізувати за допомогою теоретико-числового перетворення (NTT). Тим не менш, час множення є більше, ніж час гешування, тому множення може підходити для атак на помилки. Варто зазначити, що зменшення кількості коефіцієнтів помилки збільшує ймовірність успіху оскільки непорушені коефіцієнти w чітко проходять відхилення, а одного зміненого достатньо для досягнення $\Delta c \neq 0$. Таким чином, на відміну від інших сценаріїв з [7], таке застосування помилки має набагато сильніший вплив.

Більш імовірно, що атака буде успішною якщо вносити помилку не у прямий NTT від y, а у зворотній, застосований до w, у цьому випадку порушені лише два коефіцієнти. Однак однокоефіцієнтні помилки трохи рідше призводять до успішного відновлення ключів. Це пов'язано з тим, що коефіцієнт помилки w' може округлиться до правильного $w'_1 = w_1$, що призводить до $\Delta c = 0$ і помилку не можна використати.

4.3. Випадкова помилка при завантаженні особистого ключа та розширенні початкового значення

Сутність даної помилки – виконати розширення початкового значення ρ до матриці A. Дана операція є привабливою з трьох причин:

- помилка вноситься перед входом у цикл переривання і, таким чином, завжди виконується за один час;

- розширення до A дає головний внеск до загального часу виконання;

- A має більший слід (20 кБ у Dilithium-III), ніж інші змінні, і потенційно зберігається в пам'яті протягом тривалого часу. Тому, при обчисленні його кеш-значення не потрібно повторно запускати ExpandA для кожної операції підпису. Також існує атака на пам’ять Rowhamer, яка має схожий принцип дії.

Різниці (диференціали) у матриці A можна досягти, використовуючи початкове значення ρ, наприклад, під час завантаження особистого ключа, або вносячи помилку у розширення $A \in R_q^{k \times l} := \text{ExpandA}(\rho)$. У першому випадку поліноми $k \cdot l$, що містять A, створюються за допомогою незалежних викликів SHAKE у методі розширення ρ до A. І помилка у перестановці SHAKE призводить до лише одного полінома з помилкою. Отже, після множення матриці на вектор Ay маємо n різних коефіцієнтів w. У другому випадку, помилка вводиться напряму у ρ під час імпорту або при зберіганні значення, що призводить до зовсім іншого значення A і, таким чином до w. Цей тип помилок варто застосовувати, якщо використовуються контр заходи, такі як подвійне обчислення, тому що у цьому разі помилку складно виявити. Хоча ймовірність успіху другого методу менше у порівнянні з першим [7].

4.4. Повторне використання нонсу

Задача даної атаки – створити сценарій повторного використання нонсу з використанням помилок для генерації екземпляру LWE який буде мати тривіальне розв’язання. Спільною рисою ряду схем (NewHope, Frodo, Dilithium, Kyber) на основі LWE є процедура вибірки помилок.

Логіка атаки: У процедурі генерації ключів як компонент помилки, так і компонент секрету екземпляру LWE генеруються з використанням початкових значень (seed), які відрізняються один від одного лише одним значенням нонсу. Ці початкові значення далі вводяться до функції вибірки, яка генерує необхідні поліноми за рахунок використання функцій XOF з родини SHA3. Найбільш важливим спостереженням є те, що початкові значення відрізняються лише значенням нонсу, щоб згенерувати компоненти секрету та помилки екземпляру LWE. Спосіб виконання даної атаки полягає у тому щоб пересвідчитись, що використовуються однакові нонси як для генерації секрету, так і помилки. А робиться це за допомогою внесення відповідних помилок.

Далі показано, як знайдені вразливості можуть призвести до атак відновлення особистого ключа Dilithium.

У [9] було досліджено можливість здійснення атаки по відновлення ключа на Dilithium, на схему, яка виконує операції з модулями (матрицями/векторами/поліномами у кільці). Атака націлена на процедуру генерації ключів.

Визначимо екземпляр LWE t як тип LWE_{PK}. Кратні поліноми у відповідних компонентах секрету та помилки створюються з використанням дуже схожих початкових значень (seed), які відрізняються лише значенням нонсу, яке детерміновано збільшується. Якщо до реалізації повторного використання нонсу, при генерації компонентів секрету і помилки, можна внести кратні помилки, то $\mathbf{t}$ зводиться до множини визначених лінійних рівнянь ($k \times n$ невідомих оскільки $k > l$).

Існує достатньо специфічна але істотна відмінність щодо розкриття екземплярів LWE у схемі Dilithium. Відкритий ключ розкриває лише $\mathbf{t}_1$, d верхніх біт $\mathbf{t}$, тоді як $\mathbf{t}_0$ (компонент нижнього порядку) є частиною особистого ключа. Навіть при повторному використанні нонсу не має можливості тривіально отримати секретне значення s з помилкового відкритого ключа. Але відзначимо, що аналіз стійкості підпису Dilithium робиться за припущення, що усе $\mathbf{t}$ можна відновити лише при перегляді декількох підписів, які були генеровані на одному ключі. Таким чином доцільно припустити, що після успішного внесення помилок до

процедури генерації ключів, це призводить до успішної атаки відновлення ключа у схемі підпису Dilithium.

4.5. Часткове повторне використання нонсу

Даний спосіб атаки спершу застосовувався до підписів на еліптичній кривій [10], потім був перенесений на схеми на решітках та для Dilithium було отримано 100 % ймовірність успіху атаки [7].

Алгоритм атаки:

1) Внести помилку $\mathrm{y}' \in S^{l}_{\gamma_1-1}$, але таку, що лише один коефіцієнт $\left(\underline{y'}_u\right)_v \in \mathrm{y}$ (з індексом $u \in \{0,...,l-1\}$) та $v \in \{0,...,n-1\}$ змінюється на випадкове значення. Це все ще призводить до зовсім іншого w_1, а отже, до різного c' та $\mathrm{z}' = \mathrm{y}' + c'\mathrm{s}_1$.

2) Обчислити $\mathrm{s}_1 = \Delta c^{-1} \cdot \Delta \mathrm{z}$.

3) Визначити u, просто використовуючи один індекс для яких $s_{1,u} \notin S_\eta$. Для всіх $i \neq u$ маємо, що $\Delta y_i = 0$, таким чином вдається відновлення цих поліномів ключа.

4) Встановити поріг $\delta = 60\eta$, такий щоб $\|c\mathrm{s}_1\|_\infty \le \delta$ виконується для будь-яких c і s_1.

5) Визначити індекс v і далі положення помилки допомогою індексу максимального $|\Delta z_u|$. Оскільки введена помилка $\left(\underline{y'}_u\right)_v$ може спричинити будь-яку різницю від значення, але в середньому вона буде великою. Як очікувалось $\left|\left(\underline{y'}_u\right)_v - \left(\underline{y}_u\right)_v\right|$ буде $\frac{2_{\gamma_1}-1}{3}$, оскільки обидва ці коефіцієнти є випадковими значеннями в $\left[-(\gamma_1-1),(\gamma_1-1)\right]$ (за припущенням).

6) Відновити $s_{1,u}$. Для цього потрібно видалити рядок v лінійної системи $s_{1,u} = \Delta c^{-1} \cdot \Delta z_u$, відгадати значення $\left(\underline{s}_{1,u}\right)_v$ (вичерпним пошуком), розв'язати для повного $s_{1,u}$ і перевірити чи є він у S_η.

5. Заходи щодо протидії можливим атакам

Після аналізу ряду джерел було знайдено п’ять заходів протидії атакам на помилки. Далі наведено сутність заходів, їх переваги та недоліки.

5.1. Повторне обчислення підпису

Для того щоб виявити помилку у підписі, можна виконати його повторне обчислення та порівняти два підписи. Але якщо внести помилку двічі, або помилка є постійною (якщо внести її при завантаженні особистого ключа), то цей захід не захистить від атаки. Також його недоліком є подвійний час виконання.

5.2. Перевірка після підпису

Більшість способів атаки призводять до того, що підписи є недійсними. Таким чином, якщо перевірити підпис, то буде виявлено помилку підпису, що є ефективним заходом протидії. При чому, час перевірки приблизно у три рази менше часу вироблення підпису [11], тому цей варіант є більш ефективним ніж повторне обчислення підпису. Але недоліком є те, що даний захід не може виявити помилки, внесені до вибірки y, оскільки при цьому виробляються дійсні підписи.

5.3. Додаткова випадковість

Ще одним контрзаходом є внесення додаткової випадковості до детермінованої вибірки шуму $\mathbf{y}$. Можна просто взяти вибірку випадкового $r \leftarrow 0,1^{256}$, а потім викликати $\mathbf{y} := \text{DeterministicSample}(K \| \mu \| \kappa \| r)$. Це ефективно пом'якшує диференційну атаку на помилки, оскільки виклик помилки до алгоритму підпису використовує різні $\mathbf{y}$ і, отже $\| \mathbf{y} \neq 0$.

Більше того, цей метод також може перешкодити подальшим атакам по стороннім каналам, що стають побічним ефектом детермінізму. Як зазначають [12, 13], змішування відомого повідомлення μ з таємним початковим значенням K у геш-функції (у Dilithium це SHAKE у DeterministicSample) відкриває двері для DPA-подібних атак. Функції гешування важко захистити від таких атак; використання додаткового випадкового значення може бути альтернативою. Додатковими перевагами цього контрзаходу є його простота та незначна тривалість виконання. Крім того, на відміну від прямих реалізацій двох попередніх контрзаходів, він є однопрохідним і тому не потребує збереження копії повідомлення в пам'яті. Наприклад, автори qTESLA вимагають використовувати цей контрзахід через наявність зазначених вище атак.

Недоліком даного заходу протидії є його ймовірнісний характер, та необхідність у генераторі випадкових чисел. Такий генератор може бути доступний не на всіх пристроях, особливо малоресурсних. Також, введення цього контрзаходу скасовує гарантії стійкості, хоча конкретні атаки невідомі. Автори Dilithium "все ще рекомендують використовувати детерміновані підписи, за винятком середовищ, які можуть бути вразливими до вищезгаданих атак по стороннім каналам" [11]. Однак визначити, чи є вразливим середовище чи ні, непросто, як це чітко показано помилкою Rowhammer.

5.4. Перевірка значення нонсу

Щодо атаки повторного використання нонсів у еталонній реалізації, Dilithium може стати легкою мішенню за рахунок атак на помилки. Основна причина, однак, пов'язана з використанням початкових значень (seed) для створення компонентів секрету та помилки, які змінюються лише на один або два байти через нонс. Значення нонсу насамперед визначає різницю між компонентами секрету та помилки. Таким чином, важливо провести перевірку значення нонсу, що може унеможливити атаку. Існує маса відомих вразливостей задачі LWE, в [7] реалізували одну із вразливостей за допомогою помилок. Таким чином, проведення простих перевірок секретних та помилкових компонентів екземплярів LWE на предмет відомих тривіальних слабкостей також може бути потенційним контрзаходом проти даної атаки.

5.5. Обчислення середнього значення та дисперсії вибірки

Наступний захід протидії, запропонований у [7], обчислює середнє значення вибірки та дисперсію вибірки, одночасно перевіряючи наявність повторень. Перевірка дисперсії вибірки є особливо важливою, оскільки цей параметр пов’язаний зі складністю задачі LWE криптографічної схеми. Зменшення дисперсії вибірки помилок може спростити вирішення екземплярів LWE. Ці контрзаходи також виявляють помилки у вибірці та будь-які зловмисні реалізації. В апаратному забезпеченні цей тест буде обчислюватися після обсягу вибірки в два рази, тому не має необхідності явно обчислювати розподіл. Це зручно для такої схеми як Dilithium, яка вимагає $n = 256$ для кожного підпису.

Обчисливши середнє значення та дисперсію, можна провести статистичні тести, щоб побачити, чи ці значення є допустимими. Для середнього значення проводиться t-тест, щоб побачити, чи середнє значення цієї вибірки знаходиться в допустимих межах (попередньо

розрахованих). Для дисперсії проводиться тест хі-квадрата, щоб побачити, чи дисперсія вибірки знаходиться в межах допустимого заздалегідь визначеного значення.

Даний контрзахід додає приблизно 44 % необхідних ресурсів до вибірки CDT. Однак він має фіксовану вартість, та необхідності у додаткових апаратних ресурсах не буде, якщо дискретизатор помилок був повільнішим чи більшим. Цей контрзахід також має мінімальний вплив на продуктивність, оскільки більшість розрахунків обчислюються в паралельному розподілі помилок. В цілому для завершення обчислень потрібен лише один додатковий такт після того, як розподіл помилок завершить генерацію n вибірок.

Висновки

1. Із наведеного можна зробити висновок, що диференційні атаки на помилки становлять загрозу сучасним та перспективним схемам електронного підпису. це стосується і фіналіста конкурсу на постквантовий стандарт Dilithium.

2. Слабкими властивостями схеми Dilithium щодо атаки на помилки є геш-функція (можливо внести випадкову помилку при зверненні до геш-функції та в операцію множення поліномів), а також функція розширення початкового значення та стадія завантаження особистого ключа. Також з'являється вразливість через використання нонсу – порушник може використати двічі однакове значення нонсу (частково чи повністю).

3. Основними методами захисту від вказаних атак можуть бути:

- повторне обчислення підпису;
- перевірка підпису після підписання, що є втричі швидшим ніж попередній метод;
- внесення додаткової випадковості до детермінованої вибірки шуму;
- перевірка значення секретних та помилкових компонентів (нонсу);
- обчислення середнього значення та дисперсії вибірки, та перевірка на приналежність заданому діапазону.

Так, наприклад, зменшення дисперсії свідчить про те, що варіант LWE, на якому заснована стійкість Dilithium, значно простіше вирішити.

4. Тому розробникам постквантових схем ЕП доцільно врахувати знайдені вразливості для забезпечення стійкості до атак на помилки та застосувати знайдені рішення щодо захисту від даних атак.

5. Потрібно взяти до уваги можливі побічні ефекти після застосування контрзаходів, такі як збільшення необхідних ресурсів, порушення вимоги детермінованості схеми, зменшення ефективності.

Список літератури:

1. CRYSTALS-Dilithium. Algorithm Specifications and Supporting Documentation. Access mode https://pq-crystals.org/dilithium/data/dilithium-specification-round2.pdf

2. Відео-конференція NIST. Режим доступу: https://icmconference.org/?page_id=14324

3. Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the Importance of CheckingCryptographic Protocols for Faults (Extended Abstract). InEUROCRYPT, Lecture Notesin Computer Science, pages 37–51. Springer, 1997.

4. Eli Biham and Adi Shamir. Differential Fault Analysis of Secret Key Cryptosystems. InCRYPTO, Lecture Notes in Computer Science, pages 513–525. Springer, 1997.

5. H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan. The sorcerer's apprentice guide to fault attacks // Proceedings of the IEEE, vol. 94, no. 2, pp. 370–382, 2006.

6. Y. Kim, R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu. Flipping bits in memorywithout accessing them: An experimental study of dram disturbance errors // SIGARCH Comput. Archit. News, vol. 42,no. 3, pp. 361–372, Jun. 2014. Access mode: http://doi.acm.org/10.1145/2678373.2665726

7. Leon Groot Bruinderink and Peter Pessl. Differential Fault Attacks on Deterministic Lattice Signatures // Access mode: https://eprint.iacr.org/2018/355.pdf

8. Vadim Lyubashevsky, Chris Peikert and Oded Regev. A Toolkit for Ring-LWE Cryptography. // EUROCRYPT, volume 7881 ofLecture Notes inComputer Science, pages 35–54. Springer, 2013.

9. Prasanna Ravi1, Debapriya Basu Roy, Shivam Bhasin, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. Number "Not Used" Once – Practical fault attack on pqm4 implementations of NIST candidates. Access mode: https://eprint.iacr.org/2018/211.pdf

10. Christopher Ambrose, Joppe W. Bos, Björn Fay, Marc Joye, Manfred Lochter, and Bruce Murray. Differential Attacks on Deterministic Signatures // CT-RSA, volume 10808 of LNCS, pages 339–353. Springer, 2018.

11. Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Peter Schwabe, Gregor Seiler and Damien Stehlé // CRYSTALS-Dilithium. Submission to the NIST Post-Quantum Cryptography Standardization [NIS], 2017. Access mode: https://pq-crystals.org/dilithium.

12. Hermann Seuschek, Johann Heyszl, and Fabrizio De Santis. A Cautionary Note: Side-Channel Leakage Implications of Deterministic Signature Schemes // CS2@HiPEAC, pages 7–12. ACM, 2016.

13. Niels Samwel, Lejla Batina, Guido Bertoni, Joan Daemen, and Ruggero Susella. Breaking Ed25519 in WolfSSL // CT-RSA, volume 10808 of Lecture Notes in Computer Science, pages 1–20. Springer, 2018.

14. James Howe, Ayesha Khalidy, Marco Martinoli, Francesco Regazzoniz and Elisabeth Oswald. Fault Attack Countermeasures for Error Samplers // Lattice-Based Cryptography. Access mode: https://eprint.iacr.org/2019/206.pdf

Надійшла до редколегії 02.09.2020

Відомості про авторів:

Горбенко Юрій Іванович – канд. техн. наук, АТ «інститут інформаційних технологій», перший заступник головного конструктора, e-mail: gorbenkou@iit.kharkov.ua, ORCID: https://orcid.org/0000-0003-0073-9107

Дроздова Ольга Сергіївна – АТ «інститут інформаційних технологій», аналітик з систем захисту інформації, Україна, e-mail: 4akolzinaolga@gmail.com