УДК 004.056.5 **DOI:10.30837/rt.2023.4.215.04**

І.Д. ГОРБЕНКО, д-р техн. наук, Є.Ю. КАПТЬОЛ

АНАЛІЗ ТА ПОРІВНЯННЯ БЕЗПЕКИ ЕЛЕКТРОННИХ ПІДПИСІВ, ЩО ГРУНТУЮТЬСЯ НА НОВИХ КВАНТОВОСТІЙКИХ ПРОБЛЕМАХ

Вступ

В результаті проведення трьох раундів NIST PQC було обрано для стандартизації чотири кандидати (механізм інкапсуляції ключа CRYSTALS-Kyber та електронні підписи (ЕП) CRYSTALS-Dilithium, Falcon та SPHINCS+) та визначено кандидатів для проведення четвертого раунду (механізми інкапсуляції ключів BIKE, Classic McEliece, HQC та SIKE (котрий розробники визнали ненадійним)) [1, 2].

Через специфіку обраних алгоритмів NIST потребував додаткових кандидатів з числа ЕП загального призначення, котрі не були б засновані на використанні решіток. Через це було розпочато процес стандартизації додаткових ЕП для квантовостійкої криптографії. Серед поданих на розгляд до першого раунду цього процесу стандартизації можна виділити наступні види підписів [3, 4]: підписи, засновані на кодах; підписи на лізогеніях; мультиваріативні підписи; симетричні підписи; MPC-in-the-head та підписи, визначені NIST як "інші".

Метою роботи є аналіз та порівняння кандидатів на квантовостійкий ЕП, що грунтуються на нових та перспективних квантовостійких проблемах, стійких до класичних та квантових атак та атак бічними каналами.

Основна частина

Конкурс NIST PQC було спрямовано на обрання постквантових кандидатів криптопримітивів для стандартизації. З часом, під час розгляду та відкритого коментування проєктів стандартів криптографічною спільнотою було прийнято замінити термін «пост-квантовий» на більш точний «квантовостійкий» [5]. Саме тому в цій роботі буде використовуватись саме термін «квантовостійкий».

В межах роботи розглядаються кандидати на квантовостійкий ЕП, що були представлені на процес стандартизації додаткових ЕП від NIST.

Особливий інтерес для порівняння представляють підписи, котрі не були віднесені до жодної з груп та були об'єднані під назвою "інші підписи". Серед них наявні наступні варіанти ЕП [3]: ALTEQ, eMLE-Sig 2.0, KAZ-SIGN, Xifrat1-Sign.I, Preon.

В даній роботі розглянуто лише схеми ЕП ALTEQ, eMLE-Sig 2.0, KAZ-SIGN, Xifrat1-Sign.I.

Розглянемо схему **ЕП ALTEQ**.

Схема ЕП ALTEQ грунтується на складності проблеми рівності альтернованих трилінійних форм (ATFE), яка використовує групову дію загальної лінійної групи над скінченним полем.

Загальна структура ALTEQ полягає в наступному.

Спочатку за прикладом Голдеріх – Мікалі – Вігдерсон (GMW) розроблено протокол з нульовим розголошенням, що опирається на складність ATFE. Далі застосовується перетворення Фіат – Шаміра (FS) для усунення взаємодії від протоколу нульового розголошення, що приводить до схеми ЕП.

Протокол складається з двох частин [6]. Спочатку йде застосування протоколу GMW до рівності альтернованих трилінійних форм для отримання протоколу ідентифікації (або Сігма протоколу). Далі йде застосування перетворення Фіат – Шаміра до протоколу ідентифікації.

Базова структура GMW-FS. GMW-FS приймає групову дію і надає схему ЕП.

Групова дія, що лежить в основі ATFE [6]. Нехай G – скінченна група, S – скінченна множина, а $\alpha : G \times S \to S$ – групова дія. Припускається, що елементи цих груп та множини

ефективно представлені в алгоритмах, α може бути ефективно обчислено та елементи з G та S можуть бути ефективно відібрано випадковим чином.

Схема ALTEQ отримується шляхом інстанціювання групової дії $\alpha : G \times S \to S$ наступним чином.

Параметри для групової дії ATFE.

1. n: *розмірність векторного простору*.
2. q: *порядок кінечного поля*.

Визначення групової дії ATFE.

1. *Група* G *є* $\mathrm{GL}(n,q)$, *загальна лінійна група над скінченним полем порядку* q.
2. *Набір* S *є набором альтернуючих трилінійних форм* $\mathrm{ATF}(n,q) := \left\{\phi : \mathrm{F}_q^n \times \mathrm{F}_q^n \times \mathrm{F}_q^n \to \mathrm{F}_q\right\}$,
   де ϕ *є трилінійним* (*лінійним в кожному аргументі*)
   та альтернуючим (ϕ *прирівнюється до* 0 *коли два аргументи ідентичні*).
3. *Дія* α *визначається наступним чином. Для* $A \in \mathrm{GL}(n,q)$ *та* $\phi \in \mathrm{ATF}(n,q)$,
   $\phi \circ A$ *становить альтернуючу трилінійну форму*
   визначену $(\phi \circ A)(u,v,w) = \phi\left(A^t(u), A^t(v), A^t(w)\right)$.

Нотація. Для $n \in \mathbb{N}$, $[n] := \{1,2,...,n\}$. Нотація $\leftarrow_R$ позначає рівномірну випадкову вибірку; наприклад $g \leftarrow_R G$ позначає, що g є рівномірною випадковою виборкою з G.

Параметри для базової структури GMW-FS.

1. $C = 2^c$: Число наборів елементів в якості відкритого ключа та число груп елементів в якості секретного ключа.
2. r: Число раундів схеми.

Генерація ключів.

1. $s_1 \leftarrow_R S$.
2. $g_1 := Id$, *елемент ідентичності в групі* G.
3. $g_2,...,g_C \leftarrow_R G$
4. *Для* $i = 2,...,C, s_i := \alpha(g_i, s_1)$.
5. *Відкритий ключ* $(s_1,...,s_C) \in S^C$.
6. *Секретний ключ* $(g_1,...,g_C) \in G^C$.

Алгоритм генерації ключів ЕП ALTEQ [7] наведено на рис. 1.

Input: The variable number $n \in \mathbb{N}$, a prime power q, the alternating trilinear form number $C+1$.
Output: Public key: $C+1$ alternating trilinear forms $\phi_i \in \mathrm{ATF}(n,q)$ such that $\phi_i \cong \phi_j$ for any $i,j \in \{0,\ldots,C\}$.
Private key: C matrices $A_0,\ldots,A_{C-1}$, such that $\phi_i \circ A_i = \phi_C$.

1 Randomly sample an alternating trilinear form $\phi_C : \mathbb{F}_q^n \times \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q$.
2 Randomly sample C invertible matrices, $A_0,\ldots,A_{C-1} \in \mathrm{GL}(n,q)$.
3 For every $i \in \{0,\ldots,C-1\}$, $\phi_i \leftarrow \phi_C \circ A_i$.
4 For every $i \in \{0,\ldots,C-1\}$, $A_i \leftarrow A_i^{-1}$.
5 **return** *Public key:* $\phi_0,\phi_1,\phi_2,\ldots,\phi_C$. *Private Key:* $A_0,\ldots,A_{C-1}$.

Рис. 1. Алгоритм генерації ключів ЕП ALTEQ

Підпис. Нехай M – підписуване повідомлення. Нехай $H : \{0,1\}^* \to \{0,1\}^l$ – геш-функція, де $l = r \cdot c$.

1. *Для* $i \in [r]$, $h_i \leftarrow_R G$. *Нехай* $t_i := \alpha(h_i, s_1)$.

2. *Нехай* $L := H(M \mid t_1 \mid ... \mid t_r) \in \{0,1\}^l$.

L розділяється на r c - бітних рядків, наприклад $L = b_1 \mid ... \mid b_r$, *де* $b_i \in \{0,1\}^c$.

3. *Для* $i \in [r]$, *нехай* $f_i := h_i \cdot g_{b_i}^{-1}$.

4. *Підпис* $(b_1, ..., b_r, f_1, ..., f_r)$.

Варто зауважити, що $\alpha(f_i s_{b_i}) = \alpha(h_i \cdot g_{b_i}^{-1}, s_{b_i}) = \alpha(h_i, \alpha(g_{b_i}^{-1}, s_{b_i})) = \alpha(h_i, s_1) = t_i$.

Алгоритм генерації підпису ЕП ALTEQ [7] наведено на рис. 2.

Input: The public key $\phi_0, \ldots, \phi_C \in \mathrm{ATF}(n, q)$. The private key $A_0, \ldots, A_{C-1} \in \mathrm{GL}(n, q)$. r, C, $\lambda \in \mathbb{N}$. Let $A_C = I$, the identity matrix. The message M. A hash function $H : \{0,1\}^* \to \{0,1\}^{2\lambda}$. An expander $\mathsf{Expand} : \{0,1\}^{2\lambda} \to \{a_i\}_{i \in \{0,\ldots,r-1\}}$, where $a_i \in \{0, \ldots, C\}$.
Output: The signature S on M.

for $i \in \{0, \ldots, r-1\}$ **do**
  Randomly sample $B_i \in \mathrm{GL}(n, q)$.
  $\psi_i \leftarrow \phi_C \circ B_i$.
end
Compute $\mathsf{cha} = H(M|\psi_0| \ldots |\psi_{r-1}) \in \{0,1\}^{2\lambda}$.
$(b_0, \ldots, b_{r-1}) \leftarrow \mathsf{Expand}(\mathsf{cha})$
for $i \in \{0, \ldots, r-1\}$ **do**
  $D_i \leftarrow A_{b_i} B_i$; // Note that $\phi_{b_i} \circ D_i = \psi_i$.
end
return $S = (\mathsf{cha}, D_0, \ldots, D_{r-1})$.

Рис. 2. Алгоритм генерації підпису ЕП ALTEQ

Перевірка. Перевірювач отримує повідомлення M та підпис $(b_1, ..., b_r, f_1, ..., f_r)$.

1. *Для* $i \in [r]$, *нехай* $t_i' := \alpha(f_i, s_{b_i})$.

2. *Нехай* $L' := H(M \mid t_1' \mid ... \mid t_r')$.

3. *Прийняти якщо* L' *ідентично* $L = b_1 \mid ... \mid b_r$. *В іншому випадку відхилити.*

Алгоритм перевірки підпису ЕП ALTEQ [7] наведено на рис. 3.

Input: The public key $\phi_0, \ldots, \phi_C \in \mathrm{ATF}(n, q)$. The signature $S = (\mathsf{cha}, D_0, \ldots, D_{r-1})$, $b_i \in \{0, \ldots, C\}$, $D_i \in \mathrm{GL}(n, q)$. The message M. A hash function $H : \{0,1\}^* \to \{0,1\}^{2\lambda}$. An expander $\mathsf{Expand} : \{0,1\}^{2\lambda} \to \{a_i\}_{i \in \{0,\ldots,r-1\}}$, where $a_i \in \{0, \ldots, C\}$.
Output: "Yes" if S is a valid signature for M. "No" otherwise.

for $i \in \{0, \ldots, r-1\}$ **do**
  Compute $\psi_i' = \phi_{b_i} \circ D_i$.
end
Compute $\mathsf{cha}' = H(M|\psi_0'| \ldots |\psi_{r-1}') \in \{0,1\}^{2\lambda}$.
$(b_0', \ldots, b_{r-1}') \leftarrow \mathsf{Expand}(\mathsf{cha}')$
if *for every* $i \in \{0, \ldots, r-1\}$, $b_i = b_i'$ **then**
  return *Yes*
else
  return *No*

Рис. 3. Алгоритм перевірки підпису ЕП ALTEQ

Реалізація схеми ALTEQ включає декілька заходів для підвищення продуктивності системи.

У схемі впроваджено незбалансовані виклики. В протоколі ідентифікації GMW кількість викликів встановлюється на певне значення, так що відповідь складає випадкову матрицю,

що розгортається з короткого seed, котрий призначений для передачі (таким чином досягається скорочення підпису). З іншого боку, впровадження такого seed призводить до збільшення кількості раундів, а отже і збільшення часу підпису та перевірки.

Заявлено два варіанти підпису: Balanced з малим відкритим ключем та звичайним підписом та ShortSig з коротким підписом (розмір відкритого ключа значно більший).

Запропоновано набори параметрів для категорій безпеки NIST I та III, а також виділено орієнтовний набір параметрів для категорії безпеки NIST V. Загальносистемні параметри для схеми наведено в табл. 1 [6].

Таблиця 1

Загальносистемні параметри для схеми ЕП ALTEQ (біт)

Категорія безпеки NIST	Режим	Параметри (n, q, r, K, C)	Секретний ключ	Відкритий ключ	Підпис
I	Збалансований	$(13, 2^{32}-5, 84, 22, 7)$	128	64192	127168
	Короткий підпис	$(13, 2^{32}-5, 16, 14, 458)$	128	4191744	76224
III	Збалансований	$(20, 2^{32}-5, 201, 28, 7)$	192	255552	392000
	Короткий підпис	$(20, 2^{32}-5, 39, 20, 229)$	192	8354112	260032
V	Збалансований	$(25, 2^{32}-5, 119, 48, 8)$	256	589056	978688
	Короткий підпис	$(25, 2^{32}-5, 67, 25, 227)$	256	16707456	511264

В ході першого раунду конкурсу зі стандартизації додаткових схем ЕП було виявлено недоліки в схемі підпису ALTEQ, що призвело до розробки нової версії у відповідь на критику та наведені вектори та приклади атак.

Маркку – Джухані О. Саарінен [8] було наведено приклад атаки на підробку підпису, що базується на зниженні складності підробки за умови, що частини підпису "seed_i" та "D_i" встановлено нулями, притому, що підпис складається з трьох частин cha, seed_i, D_i. Сама атака складається із знаходження відповідного результуючого значення функції expandChallenge() в зменшеному просторі.

В результаті розробниками було додано перевірку зворотності матриць в процесі перевірки підпису та покращено швидкодію на ~2x для процедури перевірки для збалансованого варіанту та ~4x для варіанту з коротким підписом, що в поєднанні з незначним прискоренням процедури підпису та процедури генерації ключів в частини наборів параметрів не призвело до загального зниження швидкодії через додавання перевірки зворотності матриць.

Далі розглянемо **eMLE-Sig 2.0.**

eMLE-Sig 2.0 є схемою ЕП, що грунтується на новій, оптимізованій для практичного застосування версії проблеми eMLE (англ. Embedded Multilayer Equations) та базується на алгебраїчних решітках. Заявлено покращення безпеки та ефективності нової eMLE у порівнянні зі звичайною. Особливу увагу автори звернули на атаку редукції решіток, запропоновану Пенні Лоренц, яку було застосовано до попередньої версії схеми eMLE.

Сутність проблеми eMLE полягає в наступному [9]: Нехай d позначає кількість шарів в eMLE, а p – перелік з d цілих чисел, що виступають в якості модулів для кожного шару. Таким чином, для нижнього шару модулем виступає $p[0]$, для верхнього $p[d-1]$. Всі числа в p – взаємно прості, а $p[i] < p[j]$ для $0 \le i < j < d$. Нехай n – ціле число, що визначає вимірність векторів.

Приклад нового запропонованого eMLE з трьома шарами (тобто $d = 3$), де відкритими є лише $h \in \mathbb{Z}^n_{p[2]}$ та $g_l \in \mathbb{Z}^n_{p[l]} (l \in \{0,1,2\})$

$$h = g_2 \otimes x + h_1 \bmod p[2]$$

$$h_1 = \left(g_1 \otimes x + h_0 \bmod p[1]\right) + k_1 * p[1]$$

$$h_0 = \left(g_0 \otimes x \bmod p[0]\right) + k_0 * p[0]$$

Тут оператор $\otimes$ позначає результат згортки двох векторів.

Нотація [9, 10]:

– n : вимірність за замовчуванням для всіх векторів;

– d : кількість шарів в eMLE (в поданні на конкурс зафіксовано як 3);

– p : перелік з d цілих взаємно простих чисел, з $p[l]$ в якості модуля для шару l при $0 \le l \le d-1$;

– G : перелік з d векторів, з $G[l]$ використовується для побудування значення шару l;

– $x\_\max$: ціле число, що вказує на максимальні абсолютні значення елементів секретного вектору x;

– $c\_\max$: ціле число, що обмежує елементи у змагальному векторі, що використовується у підписанні та верифікації алгоритмів;

– vc : перелік, що складається з чотирьох цілих чисел, що використовується для перевірки розмірів значень при перевірці підпису;

– H : геш-функція, накшталт SHA3-256.

Алгоритм генерації ключів ЕП eMLE-Sig 2.0 [10] наведено на рис. 4.

input : $n, d, x\_max, c\_max, \mathbf{p}, \mathbf{G}$
output: $\mathbf{x}_1, \mathbf{x}_2, \mathbf{F}_1, \mathbf{F}_2, \mathbf{h}_1, \mathbf{h}_2, pkh$

while `true` **do**
  $\mathbf{x}_1 \leftarrow [-x\_max, x\_max]^n$
  $\mathbf{x}_2 \leftarrow [-x\_max, x\_max]^n$
  $sumX = \Sigma_{i=0}^{n-1}(\mathbf{x}_1[i] + \mathbf{x}_2[i]))$
  if $|sumX| < \frac{n}{2}$ **then**
    break
  end
end
while `true` **do**
  $\mathbf{h}_1, \mathbf{F}_1, sumR_1 =$ `eMLE`$(n, d, c\_max, \mathbf{p}, \mathbf{G}, \mathbf{x}_1, \mathbf{G}[1], 0)$
  $\mathbf{h}_2, \mathbf{F}_2, sumR_2 =$ `eMLE`$(n, d, c\_max, \mathbf{p}, \mathbf{G}, \mathbf{x}_2, \mathbf{G}[1], 0)$
  if $|sumR_1 + sumR_2| < n * n$ **then**
    break
  end
end
$pkh = \mathcal{H}(\mathbf{h}_1, \mathbf{h}_2)$
return $\mathbf{x}_1, \mathbf{x}_2, \mathbf{F}_1, \mathbf{F}_2, \mathbf{h}_1, \mathbf{h}_2, pkh$

Рис. 4. Алгоритм генерації ключів ЕП eMLE-Sig 2.0

Алгоритм генерації підпису ЕП eMLE-Sig 2.0 [10] наведено на рис. 5.

input : $n, d, x\_max, c\_max, \mathbf{p}, \mathbf{G}, \mathbf{vc}, \mathbf{x}_1, \mathbf{x}_2, \mathbf{F}_1, \mathbf{F}_2, pkh, m, mlen$
output: $\mathbf{u}, \mathbf{s}$

Let $sumXn$ be the sum of negative integers in $\mathbf{x}_1$ and $\mathbf{x}_2$
Let $sumXp$ be the sum of positive integers in $\mathbf{x}_1$ and $\mathbf{x}_2$
$\mathbf{c}'_1, \mathbf{c}'_2 = \texttt{hashVec}(n, c\_max, m, mlen, \texttt{null}, pkh)$
while `true` **do**
  if $sumXp > |sumXn|$ **then**
    $y\_min \leftarrow [\lfloor \frac{|sumXn| * c\_max}{10} \rfloor, \lfloor \frac{|sumXn| * c\_max}{8} \rfloor]$
    $y\_gap \leftarrow [\lfloor \frac{sumXp * c\_max}{7} \rfloor, \lfloor \frac{sumXp * c\_max}{5} \rfloor]$
  else
    $y\_min \leftarrow [\lfloor \frac{|sumXn| * c\_max}{7} \rfloor, \lfloor \frac{|sumXn| * c\_max}{5} \rfloor]$
    $y\_gap \leftarrow [\lfloor \frac{sumXp * c\_max}{10} \rfloor, \lfloor \frac{sumXp * c\_max}{8} \rfloor]$
  end
  $\mathbf{y} \leftarrow [y\_min, \lfloor \frac{n * x\_max * c\_max}{2} \rfloor - y\_gap]^n$
  $\mathbf{u}, \mathbf{F}, \_ = \texttt{eMLE}(n, d, c\_max, \mathbf{p}, \mathbf{G}, \mathbf{y}, \mathbf{c}'_1 + \mathbf{c}'_2, 1)$
  $\mathbf{c}_1, \mathbf{c}_2 = \texttt{hashVec}(n, c\_max, m, mlen, \mathbf{u}, pkh)$
  $\mathbf{s} = \mathbf{x}_1 \otimes \mathbf{c}_1 + \mathbf{x}_2 \otimes \mathbf{c}_2 + \mathbf{y}$
  $v = \texttt{check}(n, d, x\_max, c\_max, \mathbf{p}, \mathbf{G}, \mathbf{vc}, \mathbf{F}_1, \mathbf{F}_2, \mathbf{F}, \mathbf{s}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}'_1 + \mathbf{c}'_2)$
  if $v =$ `true` **then**
    break
  end
end
return $\mathbf{s}, \mathbf{u}$

Рис. 5. Алгоритм генерації підпису ЕП eMLE-Sig 2.0

Алгоритм перевірки підпису ЕП eMLE-Sig 2.0 [10] наведено на рис. 6.

input : $n, d, x\_max, c\_max, \mathbf{p}, \mathbf{G}, \mathbf{vc}, \mathbf{h}_1, \mathbf{h}_2, \mathbf{s}, \mathbf{u}, m, mlen$
output: **true** or **false**

$pkh = \mathcal{H}(\mathbf{h}_1, \mathbf{h}_2)$
$\mathbf{c}'_1, \mathbf{c}'_2 = \texttt{hashVec}(n, c\_max, m, mlen, \texttt{null}, pkh)$
$\mathbf{c}_1, \mathbf{c}_2 = \texttt{hashVec}(n, c\_max, m, mlen, \mathbf{u}, pkh)$
$v = \texttt{checkS}(n, d, x\_max, c\_max, \mathbf{vc}, \mathbf{s})$
$\mathbf{t} = \mathbf{h}_1 \otimes \mathbf{c}_1 + \mathbf{h}_2 \otimes \mathbf{c}_2 + \mathbf{u} \bmod \mathbf{p}[d-1]$
for $l = d - 1$ **to** 0 **do**
  if $l = 0$ **then**
    $\mathbf{g} = \mathbf{G}[1] \otimes (\mathbf{c}_1 + \mathbf{c}_2) \bmod \mathbf{p}[0]$
    $\mathbf{k} = \frac{\mathbf{t} - (\mathbf{G}[0] \otimes (\mathbf{s} + \mathbf{g} + \mathbf{c}'_1 + \mathbf{c}'_2) \bmod \mathbf{p}[0])}{\mathbf{p}[0]}$
    $a = \lfloor \frac{\sum_{i=0}^{n-1} \mathbf{k}[i]}{n} \rfloor$
    $\mathbf{k} = \mathbf{k} - \mathbf{1} * a$
    $v = v$ and $(\sum_{i=0}^{n-1} (\mathbf{k}[i] * \mathbf{k}[i]) \geq \mathbf{vc}[2])$ and $(\sum_{i=0}^{n-1} (\mathbf{k}[i] * \mathbf{k}[i]) \leq \mathbf{vc}[3])$
    $\mathbf{t} = \mathbf{t} - \mathbf{G}[l] \otimes (\mathbf{s} + \mathbf{g} + \mathbf{c}'_1 + \mathbf{c}'_2) \bmod \mathbf{p}[l]$
  else
    $\mathbf{t} = \mathbf{t} - \mathbf{G}[l] \otimes \mathbf{s} \bmod \mathbf{p}[l]$
  end
end
$v = v$ and $(\mathbf{t} = 0)$
return v

Рис. 6. Алгоритм перевірки підпису ЕП eMLE-Sig 2.0

Авторами заявлено, що ця версія eMLE у порівнянні зі старою версією має підвищені безпеку та ефективність за рахунок таких факторів [9]:

- рандомізація внутрішніх шарів h_1 та h_0 за рахунок рандомізованихих шумів k_1 та k_0. За рахунок використання в них більших рандомізованих цілих чисел очікуваний вектор рішень збільшується у просторі розв'язку;
- використання згортки векторів в кожному шарі дозволяє збільшити розмірність векторів (збільшити n) без збільшення розмірів підпису;
- секретний вектор x може бути зконфігурований таким чином, щоб в ньому були менші значення для зменшення розмірів підпису та підвищення стійкості.

Запропоновано набори загальносистемних параметрів для відповідних категорій безпеки NIST наведені в табл. 2 [10].

Таблиця 2

Загальносистемні параметри для схеми ЕП eMLE-Sig 2.0 (біт)

Категорія безпеки NIST	n	d	x_max /c_max	vc	p	G	Секретний ключ	Відкритий ключ	Підпис
I	64	3	4	vc64	[5,557, 67108864]	GG64	6400	3328	2240
III	96	3	4	vc96	[5,823, 268435456]	GG96	9600	5376	3648
V	128	3	4	vc128	[5,1097, 1073741824]	GG128	12800	7680	5120

В ході першого раунду конкурсу зі стандартизації додаткових схем ЕП було виявлено недоліки в схемі підпису eMLE-Sig 2.0, що призвело до доробки схеми підпису у відповідь на критику та наведені вектори та приклади атак. Незважаючи на це, було наведено приклади успішних атак на актуальну версію підпису та визнано схему недостатньо захищеною від витоку секретного ключа в підписах. Тібоучі запропонував атаку на основі цього, а Лоренц реалізував програмне забезпечення для здійснення атаки [11].

Розглянемо ЕП **KAZ-SIGN**.

Kriptografi Atasi Zarah Digital Signature (KAZ-SIGN) грунтується на математичній проблемі 2-DLP (з англ. – проблема дискретного логарифму другого порядку), яка ще потребує більш детального криптоаналізу для визначення потенційної стійкості як до класичного, так і квантового криптоаналізу [12, 13]. Ідея полягає в складності відтворення DLP (проблеми дискретного логарифму) з відомого параметру для отримання секретного параметру. KAZ-SIGN спрямований на отримання квантової стійкості з короткими ключами та підписами та високою швидкістю виконання за умови використання простої математики для отримання потенційного кандидата для легкого переходу сучасного програмного та апаратного забезпечення.

Сутність проблеми 2-DLP можна пояснити наступним чином [12]: нехай N – складене число, g – випадкове просте число з $\mathbb{Z}_N$ порядку G_g, де $G_g \approx N^{\delta}$ щонайбільше для $\delta \in (0,1)$ та $\delta \to 0$. Потрібно обрати випадкове просте число $Q \in \mathbb{Z}_{\phi(N)}$ порядку G_Q, де $G_Q \approx \phi(N)^{\varepsilon}$ для $\varepsilon \to 1$. Тобто, обрати Q великого порядку з $\mathbb{Z}_{\phi(N)}$. Таке Q має власний натуральний порядок із $\mathbb{Z}_{\phi(G_g)}$. Цей порядок буде позначено як G_{Qg}. Відношення може бути відображено як $Q^{G_{Qg}} \equiv 1 \left(\mathrm{mod}\, G_g\right)$ та $\phi(N) \equiv 0 \left(\mathrm{mod}\, G_g\right)$.

Після цього обирається випадкове ціле число $x \in \mathbb{Z}_{\phi(G_g)}$, де $x \approx \phi\left(G_g\right)$. З рівняння

$$g^{Q^x \left(\mathrm{mod}\,\phi(N)\right)} \equiv A\left(\mathrm{mod}\,N\right) \tag{1}$$

проблему дискретного логарифму (DLP) вирішено за поліноміальний час на класичному комп'ютері та отримано значення X при відсутності еквівалентності $Q^x \equiv X\left(\mathrm{mod}\,\phi\left(N\right)\right)$ та при виконанні $g^X \equiv A\left(\mathrm{mod}\,N\right)$.

2-DLP полягає в тому, що при заданих значеннях $\left(A, g, N, Q\right)$ потрібно визначити $x \in \phi\left(G_g\right)$ при $x \approx \phi\left(G_g\right)$ такому, що виконується (1).

Алгоритм генерації ключів ЕП KAZ-SIGN [13] наведено на рис. 7.

Input: System parameters $(g, n, n_{\phi(G_g)}, N, \phi(N), \phi(\phi(N)), R, G_g)$
Output: Public verification key, V, and private signing key, α

1: Choose random $\alpha \in (2^{n_{\phi(G_g)}-2}, 2^{n_{\phi(G_g)}-1})$.
2: Compute verification key, $V \equiv g^{R^\alpha \pmod{\phi(N)}} \pmod{N}$.
3: Compute the discrete logarithm $v = \mathrm{DLog}_g(V \pmod{N})$.
4: Compute $z_1 = v - R^\alpha \pmod{\phi(N)}$.
5: **if** $z_1 \equiv 0 \pmod{\phi(N)}$ **then**
6: repeat steps 1 till 4.
7: **else** continue step 9
8: **end if**
9: Compute the discrete logarithm $z_2 = \mathrm{DLog}_R(v \pmod{\phi(N)})$.
10: **if** z_2 has a solution **then**
11: repeat steps 1 till 9.
12: **else** continue step 14
13: **end if**
14: Output public verification key V and private signing key α.

Рис. 7. Алгоритм генерації ключів ЕП KAZ-SIGN

Алгоритм генерації підпису ЕП KAZ-SIGN [13] наведено на рис. 8.

Input: System parameters $(g, n, n_{\phi(G_g)}, N, \phi(N), \phi(\phi(N)), R, G_g)$, private signing key, α, and message to be signed, $m \in \mathbb{Z}_N$

Output: Signatures, (S_1, S_2), salt, σ.

1: Generate a random salt, $\sigma \in \{0,1\}^{32}$ corresponding to message, m.
2: Compute the hash value of the message, $h = H(m \| \sigma)$.
3: Choose random ephemeral prime $r \in (2^{n_{\phi(G_g)}-2}, 2^{n_{\phi(G_g)}-1})$.
4: Compute $S_0 \equiv g^{R^r \pmod{\phi(N)}} \pmod{N}$.
5: Compute the discrete logarithm $S_1 = \mathrm{DLog}_g(S_0 \pmod{N})$.
6: Compute $z_3 = S_1 - R^r \equiv 0 \pmod{\phi(N)}$.
7: **if** $z_3 = S_1 - R^r \equiv 0 \pmod{\phi(N)}$ **then**
8: Repeat steps 3 till 6.
9: **else** Continue step 11
10: **end if**
11: Compute the discrete logarithm $z_4 = \mathrm{DLog}_R(S_1 \pmod{\phi(N)})$.
12: **if** z_4 has a solution **then**
13: Repeat steps 3 till 11.
14: **else** Continue step 16
15: **end if**
16: Compute $S_2 \equiv (\alpha + h) r^{-1} \pmod{\phi(\phi(N))}$.
17: Compute the discrete logarithm $v = \mathrm{DLog}_g(V \pmod{N})$.
18: Compute the discrete logarithm $S_{2f} = \mathrm{DLog}_{S_1}(vR^h \pmod{\phi(N)})$.
19: **if** $S_2 \equiv S_{2f} \pmod{\phi(\phi(N))}$ **then**
20: Repeat steps 3 till 18
21: **else** Continue step 23.
22: **end if**
23: Compute $\alpha_F = \mathrm{DLog}_R(v \pmod{G_g})$.
24: Compute $W_0 \equiv (\alpha_F + h) S_2^{-1} \pmod{\phi(\phi(N))}$.
25: **if** W_0 does not exist **then**
26: Repeat steps 1 till 24.
27: **else** Continue 29.
28: **end if**
29: Compute $w_1 \equiv g^{S_1} \pmod{N}$.
30: Compute $w_2 \equiv g^{R^{W_0} \pmod{\phi(N)}} \pmod{N}$.
31: **if** $w_1 = w_2$ **then**
32: Repeat steps 1 till 30.
33: **else** Continue 35.
34: **end if**
35: Output signature (S_1, S_2), salt, σ and destroy r.

Рис. 8. Алгоритм генерації підпису ЕП KAZ-SIGN

Кроки 17, 18, 19 та 20 процедури підписання утворюють процедуру виявлення підробки ЕП KAZ-SIGN type-1.

Кроки 23, 24, 25, 26, 27, 28, 29, 30, 31 та 32 становлять процедуру виявлення придатності параметрів KAZ-SIGN.

Алгоритм перевірки підпису ЕП KAZ-SIGN [13] наведено на рис. 9.

Input: System parameters $(g, n, n_{\phi(G_g)}, N, \phi(N), \phi(\phi(N)), R, G_g)$, public verification key, V, message, m, signatures, (S_1, S_2) and salt corresponding to M, σ.

Output: Accept or reject

1: Compute the hash value of the message and its corresponding salt, σ to be verified, $h = H(m\|\sigma)$.
2: Compute the discrete logarithm $v = \mathrm{DLog}_g(V \pmod N)$.
3: Compute the discrete logarithm $S_{2f} = \mathrm{DLog}_{S_1}(vR^h \pmod{\phi(N)})$.
4: **if** $S_2 \equiv S_{2f} \pmod{\phi(\phi(N))}$ **then**
5: reject signature $\perp$
6: **else** continue step 9
7: **end if**
8: Compute $\alpha_F = \mathrm{DLog}_R(v \pmod{G_g})$.
9: Compute $W_0 \equiv (\alpha_F + h)S_2^{-1} \pmod{\phi(\phi(N))}$.
10: Compute $w_1 \equiv g^{S_1} \pmod N$.
11: Compute $w_2 \equiv g^{R^{W_0 \pmod{\phi(N)}}} \pmod N$.
12: **if** $w_1 = w_2$ **then**
13: reject signature $\perp$
14: **else** continue step 16
15: **end if**
16: Compute $y_1 \equiv g^{S_1^{S_2 \pmod{\phi(N)}}} \pmod N$.
17: Compute $y_2 \equiv v^{R^{h \pmod{\phi(N)}}} \pmod N$.
18: **if** $y_1 = y_2$ **then**
19: accept signature
20: **else** reject signature $\perp$
21: **end if**

Рис. 9. Алгоритм перевірки підпису ЕП KAZ-SIGN

Кроки 2, 3, 4 та 5 в процедурі перевірки утворюють процедуру виявлення підробки ЕП KAZ-SIGN type-1.

Кроки 8, 9, 10, 11, 12 та 13 утворюють процедуру виявлення підробки ЕП KAZ-SIGN type-2.

Складність вирішення 2-DLP може бути описана наступним чином [13].

Нехай $n_{\phi(G_g)} = \ell\left(\phi\left(G_g\right)\right)$. Складність отримання x становить $O\left(2^{n_{\phi(G_g)}}\right)$. За умови застосування алгоритму Гровера на квантовому комп'ютері, складність отримання x становить $O\left(2^{\frac{n_{\phi(G_g)}}{2}}\right)$. Іншими словами, так як $\phi\left(G_g\right) \approx G_g \approx N^{\delta}$, складність отримання x становить $O\left(N^{\delta}\right)$. За умови застосування алгоритму Гровера на квантовому комп'ютері, складність отримання x становить $O\left(N^{\frac{\delta}{2}}\right)$.

Запропоновані авторами набори загальносистемних параметрів для відповідних категорій безпеки NIST [12] наведено в табл. 3.

Таблиця 3

Загальносистемні параметри для схеми ЕП KAZ-SIGN (біт)

Категорія безпеки NIST	Число простих множників в P, j	Рівень безпеки, k	Довжина параметра N	Розмір ключа, (V, N)	Розмір підпису (S_1, S_2)	Розмір ключа ЕК
I	68	128	458	916	590	256
III	100	192	738	1476	930	384
V	125	256	970	1940	1220	521

В ході першого раунду конкурсу зі стандартизації додаткових схем ЕП було виявлено недоліки в схемі підпису KAZ-SIGN, що призвело до розробки чотирьох оновлень схеми підпису у відповідь на критику та наведені вектори та приклади атак (зокрема Бернштейном [14]). Незважаючи на це, було наведено приклади успішних атак з підробки підпису на актуальну версію підпису. Атака створює підпис для будь-якого бажаного повідомлення з будь-яким відкритим ключем і перевіряє, чи підпис проходить перевірку за допомогою еталонної реалізації. Стверджується, що атака працює для всіх 100 KAT у каталогах kaz1509, kaz2321 і kaz3241.

Xifrat1-Sign.I є схемою ЕП, що входить до сімейства крипто алгоритмів, що грунтуються на використанні випадково згенерованих абелевих квазігруп з 16 елементів та передбачає створення трьох шарів абелевих квазігруп зі зростаючим розміром. В рамках Xifrat1-Sign.I передбачається використання геш-функції з безпекою 256 біт та виводом 768 біт, дві половини якого оброблюються Dup-функцією.

Також автори заявляють про подвоєння безпеки проти атак "грубої сили" проти Dup-функції з 192 біт до 384 біт.

Авторами Xifrat1-Sign.I передбачається лише категорія безпеки NIST III [15].

У схемі підпису використовується геш-функція, створена на основі XOF SHAKE-256. Її початкові вихідні дані у розмірі 768 біт інтерпретуються як 12 64-бітних не підписаних цілих чисел малого порядку. Ця геш-функція позначається як $Hx_{768}(m)$.

Алгоритм генерації ключів ЕП Xifrat1-Sign.I [16] зводиться до наступного:

1. Рівномірно випадково згенерувати 3 криптограми: c, k та q.
2. Обчислити $p_1 = D(c,k)$, $p_2 = D(k,q)$,
3. Повернути відкритий ключ $pk = (c, p_1, p_2)$ та секретний ключ $sk = (c,k,q)$.

Алгоритм генерації підпису ЕП Xifrat1-Sign.I [16] зводиться до наступного:

1. Вхідні дані: m – повідомлення
2. Обчислити $h = Hx_{768}(m)$,
3. Обчислити $s = D(h,q)$,
4. Повернути s.

Алгоритм перевірки підпису ЕП Xifrat1-Sign.I [16] зводиться до наступного:

1. Вхідні дані: m – повідомлення, S – підпис
2. Обчислити $h = Hx_{768}(m)$,
3. Обчислити $t_1 = D(p_1, s)$,
4. Обчислити $t_2 = D(D(c,h), p_2)$,
5. Якщо $t_1 = t_2$ повернути [VALID]; у іншому випадку повернути [INVALID].

Доказом коректності схеми полягає в наступному:

$t_1 = D(p_1, s) = D\big(D(c,k), D(h,q)\big)$

$t_2 = D\big(D(c,h), p_2\big) = D\big(D(c,h), D(k,q)\big)$

За обмеженою комутативністю $t_1 = t_2$.

Запропоновані набори загальносистемних параметрів для категорії безпеки NIST III [15] наведено в табл. 3.

Таблиця 4

Загальносистемні параметри для схеми ЕП Xifrat1-Sign.I (біт)

Параметри	Особистий ключ	Відкритий ключ	Підпис
Xifrat1-Sign.1	3840	2304	768
Варіант з обмеженою безпекою	2560	1536	512

В ході першого раунду конкурсу зі стандартизації додаткових схем ЕП було виявлено недоліки в схемі підпису Xifrat1-Sign.I. Пенні Лоренц в офіційних коментарях [17] наведено приклад атаки, що розраховує секретний ключ з відкритого. Заявлено, що виконання атаки займає 4 хвилини на комп'ютері з 24 ядрами. Атака заснована на тому, що множення квазігруп x*y переписується як C + Ax + By, де + позначає абелеву групу, а A та B є комутативними автоморфізмами. У зв'язку з тим, що всі використані функції змішування є афінно-лінійними картами щодо +, система, що пов'язує секретний та відкритий ключі, є лінійною і може бути зведена до лінійної алгебри. Через це використана група є ізоморфною до $\mathbb{F}_2^4$, що полегшує реалізацію атаки, котра також є актуальною і для загального випадку.

На жаль, жодного рішення для цієї атаки з боку розробників не було надано.

Порівняння підписів за параметрами.

Порівняння підписів можливе за певними параметрами або критеріями. В даному випадку будемо використовувати деякі із безумовних критеріїв.

До безумовних критеріїв [18] відносяться ті критерії, виконання яких для криптопримітиву є обов'язковим, тобто безумовним. Для асиметричних криптоперетворень типу АСШ, ПІК та ЕП цілком можна вибрати однакову систему безумовних критеріїв.

До переліку безумовних критеріїв можна віднести наступні:

1) $I_{ст.}$ – рівень криптографічної стійкості з використанням безумовних критеріїв;
2) $l_{в.к.}$ – можливі довжини відкритого ключа;
3) $l_{о.к.}$ – можливі довжини особистого (секретного) ключа;
4) $l_{рез.}$ – довжина результату криптоперетворення (збитковість);
5) $T_{пр.}$ – складність (швидкість) прямого криптоперетворення;
6) $T_{зв.}$ – складність (швидкість) зворотного криптоперетворення;
7) $T_{ген.зп.}$ – складність (швидкість) генерування загальних параметрів для відповідного режиму роботи криптоперетворення (у залежності від довжин загальних параметрів та ключів);
8) $T_{ген.кл.}$ – складність (швидкість) генерування ключа (ключової пари) у залежності від режиму роботи тощо.

Із наведеного переліку в контексті даної статті будемо розглядати наступні безумовні критерії:

- можливі довжини відкритого ключа;
- можливі довжини особистого (секретного) ключа;
- довжина результату криптоперетворення (збитковість);

Порівняння цих параметрів для наведених схем ЕП наведено у табл. 5 та на рис. 10.

Таблиця 5

Розміри підпису та ключів (біт)

Параметри	ALTEQ (збалансований)			eMLE-Sig 2.0			KAZ-SIGN			Xifrat1-Sign.1
	I	III	V	I	III	V	I	III	V	III
Особистий ключ	128	192	256	6400	9600	12800	256	384	521	3840
Відкритий ключ	64192	255552	589056	3328	5376	7680	916	1476	1940	2304
Підпис	127168	392000	978688	2240	3648	5120	590	930	1220	768

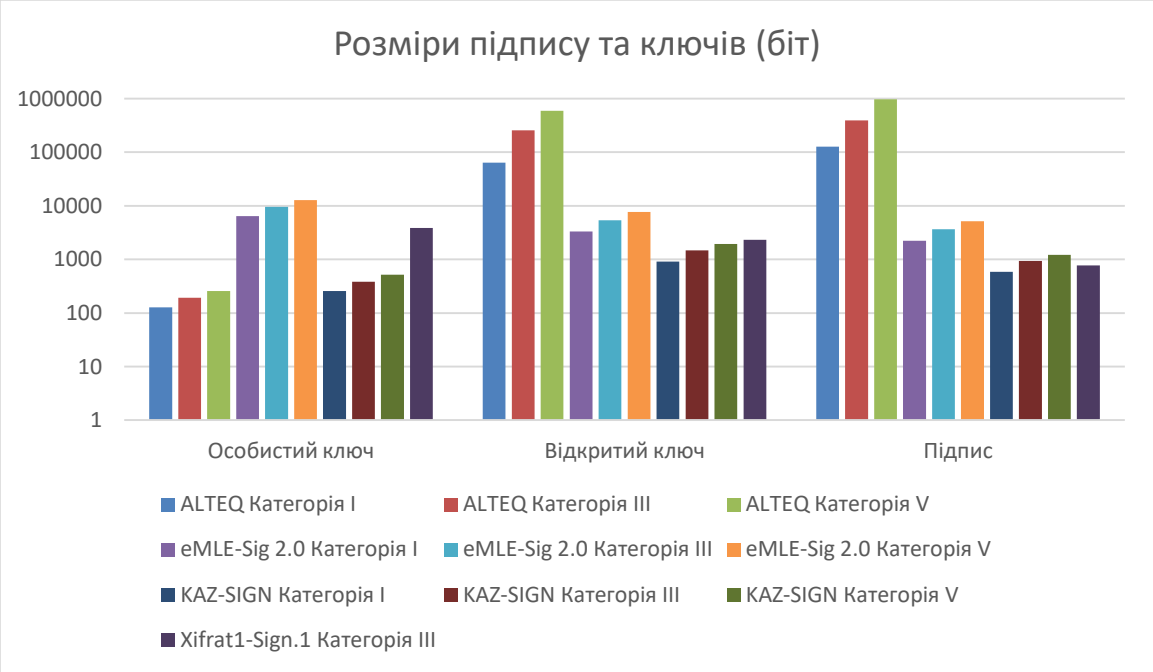

Рис. 10. Розміри підпису та ключів наведених алгоритмів ЕП

З табл. 5 та рис. 10 видно, на що робився більший наголос при розробці схем ЕП. Так, наприклад, ALTEQ явно зосереджено на зведенні до мінімуму розміру особистого ключа, хоча це і призвело до неймовірного збільшення розмірів відкритого ключа та підпису. eMLE-Sig 2.0 та Xifrat1-Sign.1 зводять до мінімуму розмір підпису. KAZ-SIGN має розміри особистого ключа, близькі до розмірів особистого ключа ALTEQ, але розміри відкритого ключа та підпису найменші з усіх порівнюваних.

Таким чином, можна побачити, що наведені схеми можуть використовуватись в різних ситуаціях. Так, наприклад, якщо дуже важливий розмір особистого ключа, але взагалі не важливі розміри відкритого ключа та підпису – цілком доцільним є використання ALTEQ. Якщо, навпаки, неважливий розмір особистого ключа, але важливі інші параметри – краще обрати іншу схему. Якщо дуже важливо звести всі розміри до мінімуму – кращим з наведених варіантів буде KAZ-SIGN.

Важливо зазначити, що таке порівняння за частиною критеріїв не є повним та не враховує затрат швидкодії та захищеність схем ЕП від конкретних атак.

Якщо розширити фокус, то можна побачити, що до кожної з наведених схем підпису було знайдено вектори атак. Додатково вартим уваги є те, що атаки бічними каналами на схеми підпису зазвичай обходять увагою, а навіть якщо на них звертають увагу, то здебільшого в контексті того, що схема підпису або цілком від них незахищена, або захищена тільки від окремих векторів атак бічними каналами, в той час як захищеність від решти навіть не розглядалась.

Висновки

Розглянуто схеми ЕП, що є кандидатами на застосування та стандартизацію в рамках процесу стандартизації додаткових ЕП для квантовостійкої криптографії від NIST. Було розглянуто основні ідеї та проблеми, що використовуються в наведених схемах ЕП для отримання квантової стійкості. Також було проведено порівняння загальносистемних параметрів для різних категорій безпеки NIST.

Так, було виявлено, що не всі кандидати орієнтовані на задоволення всіх категорій безпеки NIST. А саме ЕП Xifrat1-Sign.I передбачено лише категорію безпеки NIST III.

Також було виявлено зосередженість різних схем ЕП на зменшенні розміру різних параметрів, що призводить до переваг для різних застосувань цих схем. Таким чином, виконується одна із основних задач додаткового раунду відбору, а саме – урізноманітнення набору ЕП для стандартизації.

Окрім підписів, що можуть бути віднесені до категорій: підписи засновані на кодах; підписи на лізогеніях; мультиваріативні підписи; симетричні підписи; MPC-in-the-head; кандидати, що визначені NIST як "інші" представляють можливі підходи до квантовостійкої стійкості за рахунок використання нових, покращених та перспективних підходів та надають набори параметрів, що задовольняють вимоги NIST за різними категоріями безпеки NIST за умови використання криптографічно адекватних системних параметрів.

Список літератури:

1. Post-Quantum Cryptography PQC. Selected Algorithms 2022 : web-site. URL: https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022
2. Post-Quantum Cryptography PQC. Round 4 Submissions. URL: https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions
3. Post-Quantum Cryptography: Digital Signature Schemes. Round 1 Additional Signatures. URL: https://csrc.nist.gov/projects/pqc-dig-sig/round-1-additional-signatures
4. Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf
5. Public Comments on draft FIPS 203. Comment period: August 24, 2023 – November 22, 2023. URL: https://csrc.nist.gov/files/pubs/fips/203/ipd/docs/fips-203-initial-public-comments-2023.pdf
6. The ALTEQ Signature Scheme: Algorithm Specifications and Supporting Documentation. ALTEQ Specification Document. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/ALTEQ-Spec-web.pdf
7. Gang Tang, Dung Hoang Duong, Antoine Joux, Thomas Plantard, Youming Qiao, and Willy Susilo. Practical post-quantum signature schemes from isomorphism problems of trilinear forms. In Orr Dunkelman and Stefan Dziembowski, editors, Advances in Cryptology –EUROCRYPT 2022 – 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 - June 3, 2022, Proceedings, Part III, volume 13277 of Lecture Notes in Computer Science, pages 582–612. Springer, 2022.
8. Official Comments (Round 1 Additional Signatures) – ALTEQ. URL:https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/official-comments/ALTEQ-round1-dig-sig-official-comment.pdf
9. eMLE-Sig 2.0: A Signature Scheme based on Embedded Multilayer Equations with Heavy Layer Randomization. eMLE-Sig 2.0 Specification Document. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/eMLE-spec-web.pdf
10. Liu D. Embedded multilayer equations: a new hard problem for constructing post-quantum signatures smaller than RSA (without hardness assumption). IACR Cryptol. ePrint Arch. (2021). URL: https://eprint.iacr.org/2021/1338
11. Official Comments (Round 1 Additional Signatures) – eMLE-Sig 2.0. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/official-comments/emle-sig2.0-round1-dig-sig-official-comment.pdf
12. Kriptografi Atasi Zarah Digital Signature (KAZ-SIGN) Algorithm Specifications and Supporting Documentation. KAZ-SIGN Specification Document. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/kaz-sign-spec-web.pdf
13. KAZ-SIGN PQC Digital Signature Scheme. KAZ-SIGN NIST submissions official site. URL: https://www.antrapol.com/KAZ-SIGN
14. Official Comments (Round 1 Additional Signatures) – KAZ-SIGN. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/official-comments/KAZ-SIGN-round1-dig-sig-official-comment.pdf

15. NIST Submission: Xifrat1-Sign.I DSS. Xifrat1-Sign.I DSS Specification Document. URL:https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/xifrat1-sign-i-spec.pdf
16. Jianfang "Danny" Niu. Resurrecting Xifrat – Compact Cryptosystems 2ndAttempt. URL: https://ia.cr/2022/429
17. Official Comments (Round 1 Additional Signatures) – Xifrat1-Sign.I. URL:https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/official-comments/Xifrat1-Sign-I-round1-dig-sig-official-comment.pdf
18. Горбенко Ю. І. Науково-методичні основи аналізу, оцінки та результати порівняння існуючих та перспективних (постквантових) асиметричних криптографічних примітивів електронного підпису, протоколів асиметричного шифрування та протоколів інкапсуляції ключів / Ю. І. Горбенко, М. В. Єсіна, В. А. Пономар, І. Д. Горбенко, Є. Ю. Каптьол // Радіотехніка. 2023. Вип. 212. С. 42–65. Режим доступу: http://nbuv.gov.ua/UJRN/rvmnts_2023_212_7.

Надійшла до редколегії 15.11.2023

Відомості про авторів:
Горбенко Іван Дмитрович – д-р техн. наук, проф., Харківський національний університет ім. В.Н. Каразіна, проф. кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, АТ «Інститут Інформаційних Технологій», головний конструктор, Україна; e-mail: gorbenkoi@iit.kharkov.ua; ORCID: https://orcid.org/0000-0003-4616-3449
Каптьол Євгеній Юрійович – Харківський національний університет ім. В.Н. Каразіна, аспірант кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, АТ «Інститут Інформаційних Технологій», аналітик із систем захисту інформації, Україна; e-mail: kaptevg@gmail.com; ORCID: https://orcid.org/0000-0001-8612-2196