# МЕТОДИ ТА МЕХАНІЗМИ КРИПТОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ

**УДК 004.056.5** **DOI:10.30837/rt.2020.4.203.01**

*В. В. СЕМЕНЕЦЬ, д-р техн. наук, О. С. МАРУХНЕНКО, І. Д. ГОРБЕНКО, д-р техн. наук, Г. З. ХАЛІМОВ, д-р техн. наук*

## ПОРІВНЯЛЬНИЙ АНАЛІЗ ОДНОРАЗОВИХ ПІДПИСІВ НА БАЗІ ГЕШ-ФУНКЦІЙ

### Вступ

Стійкість значної частини сучасних асиметричних базується на складності рішення задач факторизації (RSA), дискретного логарифма в простому полі (DSA) або групі точок еліптичної кривої (ECDSA). Квантові комп'ютери, що можуть з'явитися в найближчі десятиліття, здатні вирішити ці задачі.

Актуальною є проблема аналізу та стандартизації постквантових криптосистем. З цією метою NIST проводить відкритий конкурс [1], в якому представлені алгоритми, побудовані на різних алгебраїчних структурах. В тому числі були представлені два алгоритми цифрового підпису на базі геш-функцій: Gravity SPHINCS [2] (брав участь тільки у першому раунді) та SPHINCS+[3] (успішно пройшов до третього [4]), які є незалежними модифікаціями раніше розробленого алгоритму SPHINCS [5].

Усі алгоритми сімейства SPHINCS мають складну структуру та включають декілька простіших алгоритмів підпису, серед яких одноразовий підпис Вінтерніца.

Метою статті є аналіз існуючих алгоритмів одноразового підпису, зокрема підписів Лампорта[6] та Вінтерніца[7]. Запропоновано та проаналізовано модифікації цих алгоритмів.

### 1. Огляд алгоритмів ЕЦП на базі геш-функцій

Загальна ідея підписів, що базуються на геш-функціях, полягає в тому, що у відповідність до повідомлення ставиться масив бітових рядків, елементи якого обрані та, можливо, прогешовані певну кількість разів відповідно до бітів повідомлення.

Стійкість цього класу ЕЦП базується на основних властивостях криптографічних геш-функцій – односпрямованності та стійкості до колізій. Згідно з [8] основними вимогами до функцій гешування є:

1) Складність знаходження колізії $C_{col} \geq 2^{h_{len}/2}$.

2) Складність відновлення прообразу $C_{preim} \geq 2^{h_{len}}$.

3) Складність знаходження іншого прообразу $C_{sec\_preim} \geq 2^{h_{len}}$.

Розглянемо переваги підписів на базі геш-функцій:

1) постквантовість – сучасні криптографічні геш-функції, наприклад SHA-2 та SHA-3, стійкі до квантових атак і можуть безпечно використовуватись і після створення квантових комп'ютерів;

2) легка модифікація – алгоритми ЕЦП цього класу не детермінують використовувані геш-функції, що дозволяє легко замінити механізм гешування, якщо в ньому будуть знайдені вразливості;

3) гнучкість – доповнює попередню властивість тим, що в залежності від вимог до системи можуть бути обрані геш-функції та системні параметри, що найкраще ним задовольняють, та досягається компроміс між стійкістю, швидкодією та пам'яттю.

До недоліків слід віднести:

1) обмежена кількість підписів – усі алгоритми цього класу накладають обмеження на максимальну кількість підписів, що можуть бути створені з використанням однієї пари

ключів, однак у сучасних криптосистемах ця кількість може бути дуже високою, що фактично знімає це обмеження;

2) великий розмір підпису – сучасні підписи мають складну структуру, що включає багато елементів.

ЕЦП на основі геш-функцій можна класифікувати наступним чином:

- одноразові (Лампорта [6], Вінтерніца [7 – 9]);
- багаторазові з використанням дерев Мерклі [7];
- багаторазові з поступовим зниженням стійкості (HORS[10], HORST[5], FORS[3], PORS[4]);
- багаторазові з використанням гіпердерев (SPHINCS[5], SPHINCS+[3], Gravity-SPHINCS[4], XMMS$^{MT}$[11]).

Стаття має наступну структуру: в розд. 2 наводиться опис алгоритмів ЦП Лампорта, Лампорта – Діффі та Вінтерніца, у розд. 3 описано модифіковані алгоритми, у розд. 4 проводиться порівняльний аналіз усіх описаних алгоритмів.

## 2. Алгоритми одноразового підпису

Перші алгоритми цифрового підпису на базі геш-функцій мали серйозний недолік – одна пара ключів могла бути використана для підпису тільки одного повідомлення, оскільки при підписуванні розкривалася значна частина приватного ключа і його подальше використання ставало небезпечним, ці алгоритми отримали назву одноразових підписів (One-Time Signature – OTS).

До цього класу належать:

- підпис Лампорта (LOTS);
- підпис Лампорта – Діффі (LDOTS);
- підпис Вінтерніца (WOTS);
- модифікації підпису Вінтерніца (WOTS+);
- інші варіанти.

*Зауваження 1.* В подальшому будемо розглядати узагальнений випадок, вважаючи, що для гешування повідомлення та елементів ключа використовуються різні геш-функції з довжинами вихідних значень $m$ та $n$ відповідно. На практиці, зокрема у найбільш перспективних криптосистемах сімейства SPHINCS для цього використовується одна й та сама функція з довжиною виходу $n$.

*Зауваження 2.* Далі під повідомленням мається на увазі геш-значення від фактичного повідомлення, отримане з використанням криптографічної геш-функції $H(M):\{0,1\}^{*}\to\{0,1\}^{m}$, якщо не вказано інше.

### 2.1. Підпис Лампорта (L-OTS)

Перший алгоритм використання геш-функції для створення підпису запропонований Лампортом в 1979 р. [6]. Схема була побудована наступним чином.

*Секретний ключ*. Два масиви по $m$ елементів в кожному заповнені випадковими бітовими послідовностями довжини $n$.

$$SK=\begin{pmatrix} X = (x_0, x_1, \ldots, x_{m-1}) \\ Y = (y_0, y_1, \ldots, y_{m-1}) \end{pmatrix}$$

*Відкритий ключ*. Два масиви геш-значень елементів секретного ключа.

$$PK=\begin{pmatrix} H(X) = (H(x_0), H(x_1), \ldots, H(x_{m-1})) \\ H(Y) = (H(y_0), H(y_1), \ldots, H(y_{m-1})) \end{pmatrix}$$

*Створення підпису*. Підпис – масив з $m$ значень елементів секретного ключа, обраних за таким правилом: якщо поточний біт повідомлення, що підписується, дорівнює 0, в підпис включається відповідний елемент з першого масиву секретного ключа, якщо 1 – то з другого.

Таким чином розкривається половина елементів секретного ключа, інша половина повинна бути знищена.

Перевірка підпису. Елементи підпису гешуються і порівнюються з відповідними елементами відкритого ключа.

*Зауваження 3.* Підпис Лампорта має два недоліки, що заважають його широкому використанню:

- може бути використаний для підпису тільки одного повідомлення;
- має великий розмір.

### 2.2. Підпис Лампорта-Діффі

Діффі незначно модифікував алгоритм Лампорта, уточнивши вимоги до геш-функцій, що використовуються, а саме:

- для гешування повідомлення повинна використовуватися криптографічна геш-функція;
- геш-функція, що використовується для створення відкритого ключа та перевірки підпису, повинна бути односпрямованою.

Було змінено позначення елементів секретного ключа:

$$SK = \begin{pmatrix} X_0 = (x_{0,0}, x_{0,1}, \ldots, x_{0,m-1}) \\ X_1 = (x_{1,0}, x_{1,1}, \ldots, x_{1,m-1}) \end{pmatrix}$$

Інші складові алгоритму залишилися без змін.

### 2.3. Підпис Вінтерніца (W-OTS)

З метою зменшення розмірів підпису Вінтерніц запропонував [7] використовувати ланцюгову (ітеративну) геш-функцію, що дозволяє підписувати одним значенням декілька біт повідомлення. Розглянемо цей алгоритм.

Параметр Вінтерніца – кількість біт повідомлення, що будуть одночасно підписані, в деяких джерелах параметром Вінтерніца називають значення $2^w$, будемо дотримуватися першого варіанта.

Вводяться наступні додаткові параметри:

$t_1$ – блочна довжина повідомлення;

$t_2$ – блочна довжина контрольної суми;

$t$ – блочна довжина повідомлення з контрольною сумою.

*Секретний ключ*. Масив з $t$ випадкових $n$-бітових послідовностей

$SK = (sk_0, sk_1, \ldots, sk_{t-1}), sk_i = \{0,1\}^n$.

Для економії пам'яті в якості секретного ключа можна використовувати одну випадкову послідовність $x = \{0,1\}^n$, яка при необхідності буде розгорнута у $SK$ за допомогою генератора псевдовипадкових послідовностей. Це незначно ускладнює створення підпису.

*Відкритий ключ*. Ітеративно прогешовані $(2^w - 1)$ разів елементи секретного ключа.

$PK = (pk_0, pk_1, \ldots, pk_{t-1}), \quad pk_i = f^{2^w-1}(sk_i)$.

*Створення підпису*.

Вхідні дані: M, H, SK.

Вихідні дані: $\sigma$.

1) Повідомлення розбивається на $t_1$ блоків довжиною $w$ біт $H(M) = (h_0, h_1, \ldots, h_{t_1-1})$.

2) Обчислюється контрольна сума

$$S = \sum_{i=0}^{t_1} 2^w - 1 - h_i = (s_0, s_1, \ldots, s_{t_2-1}).$$

3) Контрольна сума додається до повідомлення

$B = H(M) \,||\, S = (h_0, h_1, \ldots, h_{t_1-1}, s_0, s_1, \ldots, s_{t_2-1}) = (b_0, b_1, \ldots, b_{t-1})$, де || означає конкатенацію.

4) Обчислюється підпис

$\sigma = (\sigma_1, \sigma_0, ..., \sigma_{t-1}) = (f^{b_0}(sk_0), f^{b_1}(sk_1), ..., f^{b_{t-1}}(sk_{t-1}))$.

*Перевірка підпису.*

Вхідні дані: M, H, PK, $\sigma$.

Вихідні дані: true або false.

1) Аналогічно пунктам 1-3 алгоритму створення підпису обчислюється

$B' = (b'_0, b'_1, ..., b'_{t-1})$.

2) Обчислюється

$Z = (z_0, z_1, ..., z_{t-1}) = (f^{2^w - b_1}(\sigma_0), f^{2^w - b_0}(\sigma_1), ..., f^{2^w - b_{t-1}}(\sigma_{t-1}))$.

3) Якщо $Z = PK$, підпис коректний.

Існують різні модифікації ЕЦП Вінтерніца, зміни в основному стосуються ітеративної функції, що використовується, серед них:

- перед черговим гешуванням на число операцією XOR накладається бітова маска, це дозволяє знизити вимоги до геш-функції [9 – 11];
- використання ключової геш-функції, де результат попередньої ітерації використовується в якості ключа [8].

## 3. Модифіковані алгоритми

Розглянемо декілька модифікацій класичних алгоритмів.

### 3.1. Підпис Лампорта-Вінтерніца

В [12] запропоновано модифікацію класичних схем Лампорта та Вінтерніца, яка потребує більше детального аналізу.

Розглянемо дві варіації вдосконаленого підпису:

- Альтернативний підпис Лампорта;
- Альтернативний підпис Вінтерніца.

**Альтернативний підпис Лампорта.**

Сутність даного алгоритму полягає в зменшенні розмірів ключів та підпису за рахунок зменшення кількості біт повідомлення, що підписуються.

*Загальносистемні параметри:*

*m* – довжина геш-значення повідомлення;

*w* – кількість бітів, які підписуються одним елементом ключа;

*n* – довжина елементів ключів та підпису;

$t = \left\lceil \frac{m}{w} \right\rceil$ – кількість блоків, на які буде розбито повідомлення, а також кількість елементів ключів, що будуть для цього використані;

$H_c : \{0,1\}^* \to \{0,1\}^m$ - криптографічна геш-функція;

$H : \{0,1\}^n \to \{0,1\}^n$ – односпрямована геш-функція.

*Секретний ключ.*

Два масиви з *t* випадкових *n*-бітових рядків.

$$SK = \begin{pmatrix} X_0 = (x_{0,0}, x_{0,1}, \ldots, x_{0,t-1}) \\ X_1 = (x_{1,0}, x_{1,1}, ..., x_{1,t-1}) \end{pmatrix}$$

*Відкритий ключ.*

Два масиви геш-значень секретного ключа $SK$.

$$PK = \begin{pmatrix} H(X_0) = (H(x_{0,0}), H(x_{0,1}), \ldots, \mathrm{H}(x_{0,t-1})) \\ H(X_1) = (H(x_{1,0}), H(x_{1,1}), ..., H(x_{1,t-1})) \end{pmatrix}$$

*Створення підпису:*

Геш-повідомлення розбивається на $t$ блоків по $w$ біт (якщо $m$ не кратно $w$, у кінець додається необхідна кількість нулів).

$H(M) = (h_0, h_1, ..., h_t)$.

Відповідно до старшого біту $h_i$ в підпис додається відповідний елемент з $X_0$ чи $X_1$.

*Перевірка підпису:*

Елементи підпису гешуються та порівнюються з відповідними елементами відкритого ключа.

**Альтернативний підпис Вінтерніца.**

Сутність полягає в об'єднанні ідей Лампорта і Вінтерніца: секретний та відкритий ключі складаються з двох масивів і відкритий обчислюється з приватного за рахунок циклічного гешування. Оскільки використовується циклічне гешування для захисту від підробки використовується контрольна сума аналогічно WOTS.

Опис алгоритму.

Загальносистемні параметри:

- $H$, $f$, $w$, $t$, $t_1$, $t_2$ – аналогічно звичайному підпису Вінтерніца

*Секретний ключ.*

Два масиви з $t$–1 випадкових $n$-бітових послідовностей

$$SK = \begin{pmatrix} (sk_{0,0}, sk_{0,1}, ..., sk_{0,t-1}) \\ (sk_{1,0}, sk_{1,1}, ..., sk_{1,t-1}) \end{pmatrix}, sk_{i,j} = \{0,1\}^n .$$

*Відкритий ключ.*

Ітеративно прогешовані $2^{w-1}-1$ разів елементи секретного ключа

$$PK = \begin{pmatrix} (pk_{0,0}, pk_{0,1}, ..., pk_{0,t-1}) \\ (pk_{1,0}, pk_{1,1}, ..., pk_{1,t-1}) \end{pmatrix}, \quad pk_{i,j} = f^{2^{w-1}-1}(sk_{i,j}) .$$

*Створення підпису.*

Вхідні дані: M, H, SK.

Вихідні дані: $\sigma$.

1) Повідомлення розбивається на $t_1$ блоків довжиною $w$ біт (з метою зробити довжину повідомлення кратну $w$ до повідомлення додається необхідна кількість нульових біт)

$H(M) = (h_0, h_1, ..., h_{t_1-1})$.

2) Обчислюється контрольна сума

$$S = \sum_{i=0}^{t_1} 2^w - 1 - h_i = (s_0, s_1, ..., s_{t_2}) .$$

3) Контрольна сума додається до повідомлення

$B = H(M) \,||\, S = (h_0, h_1, ..., h_{t_1-1}, s_0, s_1, ..., s_{t_2-1}) = (b_0, b_1, ..., b_{t-1})$, де || означає конкатенацію.

4) Для кожного з блоків виділяють старший та молодші біти: $b_i = bh_i \,||\, bl_i$

5) Обчислюється підпис: в залежності від значення старшого біту обирається відповідний елемент з першої чи другої частини приватного ключа та гешується у відповідності до значення молодших біт.

$$\sigma = (\sigma_1, \sigma_0, ..., \sigma_{t-1}) = (f^{bl_0}(sk_{bh_0,0}), f^{bl_1}(sk_{bh_1,1}), ..., f^{bl_{t-1}}(sk_{bh_{t-1},t-1})) .$$

*Перевірка підпису.*

Вхідні дані: M, H, PK, $\sigma$.

Вихідні дані: true або false.

1) Аналогічно пунктам 1-3 алгоритму створення підпису обчислюється $B' = (b'_0, b'_1, ..., b'_{t-1})$.

2) Обчислюється $Z = (z_0, z_1, ..., z_{t-1}) = (f^{2^{w-1}-bl_1}(\sigma_0), f^{2^{w-1}-bl_0}(\sigma_1), ..., f^{2^{w-1}-bl_{t-1}}(\sigma_{t-1}))$.

3) Обчислюється $T = (pk_{bh'_0,0}, pk_{bh'_1,1}, ..., pk_{bh'_{t-1},t-1})$.

4) Якщо $Z = T$, підпис коректний.

### 3.2. Розширений підпис Лампорта

В класичному варіанті алгоритму Лампорта кожному біту відповідає один з двох можливих станів (елементів підключів), що призводить до значних витрат: кількість елементів підпису дорівнює кількості бітів в геш-значенні повідомлення, що підписується. Зменшення довжини геш-значення призводить до зниження стійкості. Пропонується об'єднувати біти геш-значення повідомлення в групи та ставити кожній групі бітів у відповідність по одному елементу з секретного ключа. Для цього необхідно збільшити кількість підключів до $2^w$, де $w$ – довжина групи, вибір підключа для поточного елементу здійснюється відповідно до бітового значення конкретної групи бітів. Назвемо це розширеним алгоритмом Лампорта.

*Загальносистемні параметри.*

Криптографічна геш-функція $H_c : \{0,1\}^* \to \{0,1\}^m$.

Односпрямована геш-функція $H : \{0,1\}^n \to \{0,1\}^n$.

$w$ – довжина групи бітів, яка підписується одним значенням.

*Секретний ключ.*

$2^w$ масиви по $m/w$ випадкових $n$-бітових рядків.

$SK = \left(X_0, X_1, ..., X_{2^w-1}\right)$;

$X_i = (x_{i,0}, x_{i,1}, \ldots, x_{i,(m-1)/w})$.

*Відкритий ключ.*

$2^w$ масиви по $m / w$ геш-значень елементів секретного ключа.

$PK = H(SK) = \left((H(X_0), H(X_1), \ldots, H(X_{2^w-1}))\right)$;

$H(X_i) = (H(x_{i,0}), H(x_{i,1}), \ldots, H(x_{i,(m-1)/w}))$.

*Створення підпису.*

Вхідні дані: SK.

Вихідні дані: $\sigma$.

Підпис – масив з $m / w$ значень секретного ключа, які обираються відповідно до пар бітів повідомлення.

$H_c(M) = \{h_0, h_1, ..., h_{(m-1)/w}\}, \ h_i = \{0,1\}^w$;

$\sigma = \left(x_{h_0,0}, x_{h_1,1}, ..., x_{h_{(m-1)/w},(m-1)/w}\right)$.

*Перевірка підпису.*

Вхідні дані: PK, $\sigma$.

Вихідні дані: true або false.

Елементи підпису гешуються і порівнюються з відповідними елементами відкритого ключа.

$H'_c(M) = \{h'_0, h'_1, ..., h'_{(m-1)/w}\}$;

$H(\sigma) = \left(H(x_{h_0,0}), H(x_{h_1,1}), ..., H(x_{h_{(m-1)/w},(m-1)/w})\right)$;

$Z = \left(H(x_{h'_0,0}), H(x_{h'_1,1}), ..., H(x_{h'_{(m-1)/w},(m-1)/w})\right)$.

Якщо $H(\sigma) = Z$ підпис коректний.

### 4. Аналіз параметрів одноразових підписів

До основних параметрів, за якими можливо порівнювати алгоритми ЕЦП незалежно від їх математичної бази відносяться:

- стійкість до криптоаналізу;
- розміри ключів та підписів;
- обчислювальна складність генерації ключів, створення та перевірки підпису.

У контексті одноразових підписів узагальнено можна виділити два класи атак:

- при відомому публічному ключі;
- при відомому публічному ключі та підписі.

*Зауваження 4.* Більш складні атаки, зокрема із вибором або адаптивним вибором повідомлення, розглядатися не будуть, оскільки наявніть декількох підписів, створених із використанням однієї ключової пари є неможливою за визначенням одноразового підпису. Також виникнення такої пари підписів свідчить про порушення у роботі схеми підпису і призводить до миттєвого зниження стійкості системи.

До другого класу належить атака типу «екзестенційна підробка», що може бути застосовано до довільного алгоритму підпису, а саме знаходження другого прообразу до геш-значення підписаного повідомлення. При використанні криптографічної геш-функції її складність складає $m$ біт та не може бути зменшена за рахунок накопичення аналітиком великої кількості повідомлень, оскільки, за визначенням, ключі одноразового підпису використовуються лише один раз. З іншого боку особливості підписів на основі геш-функцій дозволяють зменшити складність пошуку підходящого повідомлення, геш якого співпадає з підписаним лише частково. Детальний аналіз стосовно кожного з алгоритмів наводиться далі.

#### 4.1. Аналіз підпису Лампорта

1) Стійкість.

а) Атака при відомому відкритому ключі.

Для успішного втілення подібної атаки зловмиснику необхідно знайти перший чи другий прообраз для $m$ $n$-бітних геш-значень (можливістю коллізій між елементами приватного ключа знехтуємо, що для його генерації використовувався генератор з необхідними криптографічними властивостями). Якщо використовується криптографічна геш-функція, то складність такої атаки складає $2^{m*n}$, що відповідає стійкості $m*n$ біт. Наприклад, якщо $m = n = 256$, стійкість алгоритму складає 65536 біт.

б) Атака при відомому відкритому ключі та підписі.

Зловмисник може спробувати знизити складність атаки «екзистенційна підробка», а саме пошуку другого прообразу для підписаного повідомлення, підібравши повідомлення, геш якого співпадає з підписаним лише частково, наприклад, співпадають $(m–k)$ біт. Однак в цьому випадку йому буде необхідно знайти прообраз (перший чи другий) для кожного невідомого елементу відкритого ключа, тобто зниження складності пошуку повідомлення на 1 біт призводить до збільшення складності створення підробного підпису на $n$ біт. Складність такої атаки буде становити $(m – k + n*k = m + (n – 1)*k)$ біт.

Таким чином, стійкість криптоалгоритму складає не менше $m$ біт. Якщо невикористані елементи секретного ключа не були знищенні під час створення підпису і частина з них стала відома зловмиснику, складність підробки зменшується. Кожен зкомпрометований елемент фактично знижує ентропію геш-значення повідомлення на 1 біт, якщо криптоаналітику відомі $k$ додаткових елементів секретного ключа, то складність підробки складає $(m–k)$ біт.

2) Розміри підпису та ключа.

Довжина секретного та відкритого ключів у схемі підпису Лампорта визначається як $2*m*n$ біт, довжина ЕП $m*n$ біт. Результати оцінки розмірів секретних та відкритих одноразових ключів та розмірів ЕП для механізмів Лампорта та Лампорта – Діффі в залежності від параметрів безпеки наведені в табл. 1.

Таблиця 1

Розміри секретних та відкритих ключів
та ЕП для механізму Лампорта в кілобайтах

| Параметр безпеки ($n$) | Розмір секретного ключа | Розмір відкритого ключа | Розмір ЕП |
|---|---|---|---|
| 128 | 4 | 4 | 2 |
| 192 | 9 | 9 | 4.5 |
| 256 | 16 | 16 | 8 |
| 384 | 36 | 36 | 18 |
| 512 | 64 | 64 | 32 |

Розміри підпису Лампорта становлять від одиниць до десятків кілобайт в залежності від рівня стійкості.

3) Обчислювальна складність.

Створення пари ключів підпису Лампорта вимагає генерації випадкових значень та їх гешування, створення підпису вимагає тільки відбору необхідних значень, перевірка підпису вимагає тільки гешування. У випадку, якщо підписувач за якихось причин не зберігає публічний ключ, для створення підпису йому необхідно прогешувати відповідні елементи приватного ключа. Обчислювальну складнсть механізму наведено в табл. 2.

Таблиця 2

Обчислювальна складність механізму Лампорта

| Параметр | Генерація ключів | Створення підпису | Перевірка |
|---|---|---|---|
| Генерація випадкового значення | $2m$ | - | - |
| Обчислення геш-функції | $2m$ | $-/m$ | $m$ |

Усе зазначене також актуально для підпису Лампорта – Діффі, що фактично відрізняється тільки позначеннями та вимогами до геш-функцій.

### 4.2. Аналіз підпису Вінтерніца

Детальний аналіз стійкості підпису Вінтерніца та його модифікацій проведено у роботах [8, 9, 13]. В даній роботі ми будемо використовувати спрощену модель.

1) Стійкість.

а) Атака при відомому відкритому ключі.

Розглянемо три можливі для зловмисника випадки: найгірший, звичайний та найгірший, складність пошуку повідомлення, геш якого має необхідні властивості, не враховуємо.

В найгрішному випадку геш-значення підробного повідомлення містить тільки нульові біти, в такому випадку зловмиснику необхідно знайти прообрази для декількоразового гешування $t_1$ елементів (ймовірністю колізій знехтуємо), тобто стійкість складає не менше $n * t_1$ біт.

У звичайному (найбільш ймовірному) випадку значення груп пікселів будуть рівномірно розподілені в діапазоні $[0;2^w–1]$, тобто максимальне значення $2^w–1$, якому відповідає елемент відкритого ключа приймуть близько $(t_1/(2^w–1))$. Складність підробки при цьому складає не менше $(t_1 – t_1/(2^w–1))*n$ біт, а з урахуванням елементів, що відповідають контрольній сумі, стійкість стає вищою.

В найкращому для зловмисника випадку геш-значення підробного повідомлення містить тільки одиничні біти, в такому випадку $t_1$ елементів підпису дорівнюють елементам відкритого ключа, що відомий зловмиснику. Однак залишаються $t_2$ елементів, що відповідають контрольній сумі, яка в цьому випадку дорівнює нулю. Таким чином, зловмиснику необхідно знайти прообрази для декількоразового гешування $t_2$ елементів. Стійкість до цієї атаки дорівнює не менше $t_2*n$ біт.

б) Атака при відомому відкритому ключі та підписі.

Зловмисник може знизити складність пошуку другого прообразу, підібравши повідомлення, геш якого співпадає з підписаним лише частково. Якщо зловмисник має коректний підпис, то через певну кількість обчислень він зможе створити повідомлення, з геш-значенням $H(M') = (h'_0, h'_1, ..., h'_{t_1-1})$, таким що $h'_i \geq h_i$, таким чином після додаткових ітерацій початковий підпис стане коректним для повідомлення $M'$. Використання контрольної суми дозволяє захистити підпис від підробки: при збільшені $h_i$ значення контрольної суми зменшується, отже елементи ключа, використані для підписування контрольної суми, повинні бути прогешовані меншу кількість разів, ніж у початковому повідомленні, що зловмисник зробити не може через односторонню природу функції *f*. Таким чином, зловмиснику необхідно знайти прообраз хоча б одного *n*-бітного значення, що призводить до стійкості не менше *n* біт.

2) Розміри підпису та ключа.

Розмір підпису в схемі Вінтерніца дорівнює розміру відкритого ключа та складає *SignSize = t * n* біт. Нехай до повідомлення та елементів ключа застосовується одна й та ж сама криптографічна геш-функція з довжиною геш-значення $n = \{128; 192; 256; 384; 512\}$ біт, параметр Вінтерніца $w = [2, 3, ..., 16]$. Залежність розмірів підпису від *n* та *w* наведено у табл. 3.

Таблиця 3
Залежність розміру підпису Вінтерніца
від *n* та *w* в кілобайтах

| *N* / *w* | 128 | 192 | 256 | 384 | 512 |
|---|---|---|---|---|---|
| 2 | 1.063 | 2.367 | 4.156 | 9.234 | 16.313 |
| 3 | 0.719 | 1.570 | 2.813 | 6.188 | 10.938 |
| 4 | 0.547 | 1.195 | 2.094 | 4.641 | 8.188 |
| 5 | 0.438 | 0.984 | 1.719 | 3.750 | 6.625 |
| 6 | 0.375 | 0.797 | 1.406 | 3.094 | 5.563 |
| 7 | 0.328 | 0.703 | 1.219 | 2.672 | 4.750 |
| 8 | 0.281 | 0.609 | 1.063 | 2.344 | 4.125 |
| 9 | 0.266 | 0.563 | 0.969 | 2.109 | 3.688 |
| 10 | 0.234 | 0.516 | 0.875 | 1.922 | 3.375 |
| 11 | 0.219 | 0.469 | 0.813 | 1.734 | 3.063 |
| 12 | 0.203 | 0.422 | 0.750 | 1.594 | 2.813 |
| 13 | 0.188 | 0.398 | 0.688 | 1.500 | 2.625 |
| 14 | 0.188 | 0.375 | 0.656 | 1.406 | 2.438 |
| 15 | 0.172 | 0.352 | 0.625 | 1.313 | 2.313 |
| 16 | 0.156 | 0.328 | 0.563 | 1.219 | 2.125 |

3) Обчислювальна складність.

Для генерації приватного ключа підпису Вінтерніца необхідно згенерувати *t* випадкових *n*-бітних значень, для обчислення публічного ключа необхідно виконати $N = t * (2^w - 1)$ операцій гешування. Створення та перевірка підпису разом фактично повторюють процедуру обчислення публічного ключа, однак обчислювальна складність виконання кожної з цих процедур окремо залежить від повідомлення, що підписується. Залежність кількості гешувань для обчислення відкритого ключа схеми Вінтерніца від *n* та *w* наведено у табл. 4.

*Зауваження 5.* З таблиці видно, що на практиці можна використовувати лише невеликі значення параметра Вінтерніца, оскільки кількість ітерацій для генерації відкритого ключа, створення та перевірки підпису експоненційно залежить від даного параметру. Оптимальним, на наш погляд, є значення *w* = 4, розмір підпису в такому випадку складає 0,5 – 8 кілобайт та потребує близько 500 – 2000 операцій гешування, в залежності від рівня стійкості, що є прийнятною величиною. Також розбиття даних, що підписуються, на 4-бітні блоки легко реалізується як програмно, так і апаратно.

Таблиця 4

Залежність кількості гешувань для обчислення відкритого ключа підпису Вінтерніца від *n* та *w*

| *N* \ *w* | 128 | 192 | 256 | 384 | 512 |
|---|---|---|---|---|---|
| 2 | 204 | 303 | 399 | 591 | 783 |
| 3 | 322 | 469 | 630 | 924 | 1225 |
| 4 | 525 | 765 | 1005 | 1485 | 1965 |
| 5 | 868 | 1302 | 1705 | 2480 | 3286 |
| 6 | 1512 | 2142 | 2835 | 4158 | 5607 |
| 7 | 2667 | 3810 | 4953 | 7239 | 9652 |
| 8 | 4590 | 6630 | 8670 | 12750 | 16830 |
| 9 | 8687 | 12264 | 15841 | 22995 | 30149 |
| 10 | 15345 | 22506 | 28644 | 41943 | 55242 |
| 11 | 28658 | 40940 | 53222 | 75739 | 100303 |
| 12 | 53235 | 73710 | 98280 | 139230 | 184275 |
| 13 | 98292 | 139247 | 180202 | 262112 | 344022 |
| 14 | 196596 | 262128 | 344043 | 491490 | 638937 |
| 15 | 360437 | 491505 | 655340 | 917476 | 1212379 |
| 16 | 655350 | 917490 | 1179630 | 1703910 | 2228190 |

### 4.3. Аналіз альтернативного підпису Лампорта

Вдосконалений алгоритм відрізняється від класичного підпису Лампорта тим, що підписуються не всі біти геш-значення, а лише кожен *w*-й, що є еквівалентним до використання у підписі геш-функції з більш коротким вихідним значенням.

1) Стійкість.

Для цього алгоритму справедливі ті ж самі припущення, що і для стандартного підпису Лампорта.

а) Атака при відомому відкритому ключі.

В цьому випаку зловмиснику необхідно *t* разів знайти прообраз для *n*-бітової геш-функції, що за умови стійкості до пошуку прообразів еквівалентно вгадуванню *t*n* випадкових біт.

б) Атака при відомому відкритому ключі та підписі.

Для класичної схеми Лампорта складність пошуку другого прообразу дорівнює $2^n$, в модифікованій схемі при підписі враховуються лише *t* бітів геш-значення, відповідно складність пошуку повідомлення, що дасть геш зі співпадаючими відповідними бітами, дорівнює $2^t$.

Криптоаналітик може спростити задачу пошуку повідомлення, яке має відповідний геш, дозволивши відмінності у «критичних» *t* бітах, при цьому, як і у звичайному підписі Лампорта, відмінність у одному біті вимагає пошуку прообразу для одного з елементів відкритого ключа, складність цієї задачі $2^n$. Стійкість при ігноруванні *k* критичних біт складатиме $t + (n - 1)*k$.

*Зауваження 6.*

1. Найнебезпечнішою є друга атака, яка унеможливлює використання великих значень *w*.

2. Доцільність використання альтернативного підпису Лампорта викликає сумніви через його уразливість.

2) Розміри підпису та ключа.

Альтернативний підпис Лампорта повністю еквівалентний звичайному підпису Лампорта з довжиною геш-значення повідомлення *t* біт.

Розміри приватного та публічного ключів становлять 2**t***n* біт. Розміри підпису *t***n* біт.

3) Обчислювальна складність.

Аналогічно стандартному агоритму Вінтерніца генерація ключа включає генерацію 2$t$ випадкових значень та їх гешування. Створення та перевірка підпису вимагає t гешувань кожне.

### 4.4. Аналіз альтернативного підпису Вінтерніца

1) Стійкість

Властивості альтернативний підпису Вінтерніца в цілому схожі на властивості звичайного підпису Вінтерніца з деякими уточненнями.

а) Атака при відомому відкритому ключі.

Аналогічно звичайному підпису Вінтерніца можна виділити декілька випадків складності підробки відносно бітів геш-значення підробного повідомлення – усі біти 0, біти рівномірно розподілені, усі біти 1. Як і у стандартній схемі, стійкість до підробки у найгіршому випадку складає $n * t_1$ біт, у найкращому $n * t_2$ біт. Оскільки обидві частини відкритого ключа відомі зловмиснику, він може обирати довільний елемент у кожній парі, відповідно до повідомлення, що підписується.

б) Атака при відомому відкритому ключі та підписі.

Аналогічно механізму Лампорта криптоаналітик може спростити пошук другого прообразу для геш-значення повідомлення, ігноруючи відмінності у бітах, що визначають вибір елементу з першої чи другої підмножини. В цьому випадку для створення коректного підпису необхідно знайти невідомі елементи приватного ключа. Знаходження $k$ елементів приватного ключа означає знаходження прообразів для $k$ значень ітеративної геш-функції.

Аналогічно механізму Вінтерніца можна підібрати повідомлення, усі блоки геш-значення якого будуть більше за блоки підписаного повідомлення, як було зазначено раніше, від цього захищає використання контрольної суми. Тобто стійкість складає не менше $n$ біт.

Таким чином, зміна одного біту геш-значення у частині блоку, що відповідає за Лампортову складову, або у частині, що відповідає за складову Вінтерніца, неодмінно вимагає знаходження хоча б одного прообразу для геш-функції і стійкість залишається не нижче $n$ біт.

2) Розміри підпису та ключа.

Розмір підпису альтернативного алгоритму Внітерніца залишився незмінним і складає $SignSize = t * n$. Розміри ключів збільшилися вдвічі і дорівнюють $KeySize = 2 * t * n$.

Залежність розмірів ключів від $n$ та $w$ наведено у табл. 5.

Таблиця 5

Залежність розміру ключів (в кілобайтах) альтернативного підпису Вінтерніца від $n$ та $w$

| $N$ \ $w$ | 128 | 192 | 256 | 384 | 512 |
|---|---|---|---|---|---|
| 2 | 2.125 | 4.734 | 8.313 | 18.469 | 32.625 |
| 3 | 1.438 | 3.141 | 5.625 | 12.375 | 21.875 |
| 4 | 1.094 | 2.391 | 4.188 | 9.281 | 16.375 |
| 5 | 0.875 | 1.969 | 3.438 | 7.500 | 13.250 |
| 6 | 0.750 | 1.594 | 2.813 | 6.188 | 11.125 |
| 7 | 0.656 | 1.406 | 2.438 | 5.344 | 9.500 |
| 8 | 0.563 | 1.219 | 2.125 | 4.688 | 8.250 |
| 9 | 0.531 | 1.125 | 1.938 | 4.219 | 7.375 |
| 10 | 0.469 | 1.031 | 1.750 | 3.844 | 6.750 |
| 11 | 0.438 | 0.938 | 1.625 | 3.469 | 6.125 |
| 12 | 0.406 | 0.844 | 1.500 | 3.188 | 5.625 |
| 13 | 0.375 | 0.797 | 1.375 | 3.000 | 5.250 |
| 14 | 0.375 | 0.750 | 1.313 | 2.813 | 4.875 |
| 15 | 0.344 | 0.703 | 1.250 | 2.625 | 4.625 |
| 16 | 0.313 | 0.656 | 1.125 | 2.438 | 4.250 |

3) Обчислювальна складність.

Кількість ітерацій ланцюгової функції для обчислення публічного ключа в альтернативному підписі Вінтерніца $N_{key} = N_{key\_WOTS} - t$. Кількість ітерацій ланцюгової

функції для створення та перевірки підпису $N_{sign} = (N_{sign\_WOTS} - t) / 2$. Залежність кількості гешувань для створення та перевірки підпису та обчислення відкритого ключа від $n$ та $w$ наведено у табл. 6, 7.

Таблиця 6

Залежність кількості гешувань для підпису від $n$ та $w$

| $N$ / $w$ | 128 | 192 | 256 | 384 | 512 |
|---|---|---|---|---|---|
| 2 | 68 | 101 | 133 | 197 | 261 |
| 3 | 138 | 201 | 270 | 396 | 525 |
| 4 | 245 | 357 | 469 | 693 | 917 |
| 5 | 420 | 630 | 825 | 1200 | 1590 |
| 6 | 744 | 1054 | 1395 | 2046 | 2759 |
| 7 | 1323 | 1890 | 2457 | 3591 | 4788 |
| 8 | 2286 | 3302 | 4318 | 6350 | 8382 |
| 9 | 4335 | 6120 | 7905 | 11475 | 15045 |
| 10 | 7665 | 11242 | 14308 | 20951 | 27594 |
| 11 | 14322 | 20460 | 26598 | 37851 | 50127 |
| 12 | 26611 | 36846 | 49128 | 69598 | 92115 |
| 13 | 49140 | 69615 | 90090 | 131040 | 171990 |
| 14 | 98292 | 131056 | 172011 | 245730 | 319449 |
| 15 | 180213 | 245745 | 327660 | 458724 | 606171 |
| 16 | 327670 | 458738 | 589806 | 851942 | 1114078 |

Таблиця 7

Залежність кількості гешувань для обчислення відкритого ключа від $n$ та $w$

| $N$ / $w$ | 128 | 192 | 256 | 384 | 512 |
|---|---|---|---|---|---|
| 2 | 136 | 202 | 266 | 394 | 522 |
| 3 | 276 | 402 | 540 | 792 | 1050 |
| 4 | 490 | 714 | 938 | 1386 | 1834 |
| 5 | 840 | 1260 | 1650 | 2400 | 3180 |
| 6 | 1488 | 2108 | 2790 | 4092 | 5518 |
| 7 | 2646 | 3780 | 4914 | 7182 | 9576 |
| 8 | 4572 | 6604 | 8636 | 12700 | 16764 |
| 9 | 8670 | 12240 | 15810 | 22950 | 30090 |
| 10 | 15330 | 22484 | 28616 | 41902 | 55188 |
| 11 | 28644 | 40920 | 53196 | 75702 | 100254 |
| 12 | 53222 | 73692 | 98256 | 139196 | 184230 |
| 13 | 98280 | 139230 | 180180 | 262080 | 343980 |
| 14 | 196584 | 262112 | 344022 | 491460 | 638898 |
| 15 | 360426 | 491490 | 655320 | 917448 | 1212342 |
| 16 | 655340 | 917476 | 1179612 | 1703884 | 2228156 |

*Зауваження 7.*

1. Вдосконалений алгоритм потребує менше обчислень для створення та перевірки підпису, однак більше пам’яті для зберігання секретного та відкритого ключів.

2. Оскільки відкритий ключ неможливо повністю обчислити з підпису, цей ЕЦП не так зручно використовувати в схемах з геш-деревами – разом з підписом необхідно передавати іншу половину елементів відкритого ключа, що подвоює реальний розмір підпису.

### 4.5. Аналіз розширеного підпису Лампорта

1) Стійкість.

а) Атака при відомому відкритому ключі.

Аналогічно звичайному підпису Лампорта, для підробки розширеного алгоритму необхідно знайти прообрази для усіх елементів відкритого ключа, що будуть використані. Стійкість дорівнює $m / w * n$ біт.

б) Атака при відомому відкритому ключі та підписі.

Так само, як і для класичної схеми, криптоаналітик повинен або виконати атаку типу «єкзистенційна підробка» зі складністю $2^m$, або віднайти прообрази до елементів ключа, що відповідають зміненим бітам.

2) Розміри ключів та підпису.

Довжина секретного та відкритого ключів у модифікованому алгоритмі Лампорта визначається як $2^w * (m / w) * n$, довжина ЕП $(m / w) * n$. Результати оцінки розмірів секретних та відкритих одноразових ключів та розмірів ЕП для розширеного алгоритму Лампорта в залежності від параметрів безпеки наведені в табл. 8.

Таблиця 8

Залежність розміру ключів та підписів (в кілобайтах) розширеного підпису Лампорта від *n* та *w*

| | Ключ | | | | | Підпис | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| *N* \ *w* | 128 | 192 | 256 | 384 | 512 | 128 | 192 | 256 | 384 | 512 |
| 2 | 4 | 9 | 16 | 36 | 64 | 1,00 | 2,25 | 4,00 | 9,00 | 16,00 |
| 3 | 5,333 | 12 | 21,333 | 48 | 85,333 | 0,67 | 1,50 | 2,67 | 6,00 | 10,67 |
| 4 | 8 | 18 | 32 | 72 | 128 | 0,50 | 1,13 | 2,00 | 4,50 | 8,00 |
| 5 | 12,8 | 28,8 | 51,2 | 115,2 | 204,8 | 0,40 | 0,90 | 1,60 | 3,60 | 6,40 |
| 6 | 21,333 | 48 | 85,333 | 192 | 341,333 | 0,33 | 0,75 | 1,33 | 3,00 | 5,33 |
| 7 | 36,571 | 82,286 | 146,286 | 329,143 | 585,143 | 0,29 | 0,64 | 1,14 | 2,57 | 4,57 |
| 8 | 64 | 144 | 256 | 576 | 1024 | 0,25 | 0,56 | 1,00 | 2,25 | 4,00 |
| 9 | 113,778 | 256 | 455,111 | 1024 | 1820,44 | 0,22 | 0,50 | 0,89 | 2,00 | 3,56 |
| 10 | 204,8 | 460,8 | 819,2 | 1843,2 | 3276,8 | 0,20 | 0,45 | 0,80 | 1,80 | 3,20 |
| 11 | 372,364 | 837,818 | 1489,45 | 3351,27 | 5957,82 | 0,18 | 0,41 | 0,73 | 1,64 | 2,91 |
| 12 | 682,667 | 1536 | 2730,67 | 6144 | 10922,7 | 0,17 | 0,38 | 0,67 | 1,50 | 2,67 |
| 13 | 1260,31 | 2835,69 | 5041,23 | 11342,8 | 20164,9 | 0,15 | 0,35 | 0,62 | 1,38 | 2,46 |
| 14 | 2340,57 | 5266,29 | 9362,29 | 21065,1 | 37449,1 | 0,14 | 0,32 | 0,57 | 1,29 | 2,29 |
| 15 | 4369,07 | 9830,4 | 17476,3 | 39321,6 | 69905,1 | 0,13 | 0,30 | 0,53 | 1,20 | 2,13 |
| 16 | 8192 | 18432 | 32768 | 73728 | 131072 | 0,13 | 0,28 | 0,50 | 1,13 | 2,00 |

3) Обчислювальна складність.

Генерація приватного ключа розширеної схеми Лампорта еквівалентна генерації $2^m * m / w$ *n*-бітних значень. Для обчислення відкритого ключа необхідно прогешувати кожен з елементів приватного ключа. У табл. 9 наведена залежність кількості гешувань для обчислення відкритого ключа від *n* та *w*.

Таблиця 9

Залежність кількості гешувань для обчислення відкритого ключа від *n* та *w*

| *N* \ *w* | 128 | 192 | 256 | 384 | 512 |
|---|---|---|---|---|---|
| 2 | 256 | 384 | 512 | 768 | 1024 |
| 3 | 342 | 512 | 683 | 1024 | 1366 |
| 4 | 512 | 768 | 1024 | 1536 | 2048 |
| 5 | 820 | 1229 | 1639 | 2458 | 3277 |
| 6 | 1366 | 2048 | 2731 | 4096 | 5462 |
| 7 | 2341 | 3511 | 4682 | 7022 | 9363 |
| 8 | 4096 | 6144 | 8192 | 12288 | 16384 |
| 9 | 7282 | 10923 | 14564 | 21846 | 29128 |
| 10 | 13108 | 19661 | 26215 | 39322 | 52429 |
| 11 | 23832 | 35747 | 47663 | 71494 | 95326 |
| 12 | 43691 | 65536 | 87382 | 131072 | 174763 |
| 13 | 80660 | 120990 | 161320 | 241980 | 322639 |
| 14 | 149797 | 224695 | 299594 | 449390 | 599187 |
| 15 | 279621 | 419431 | 559241 | 838861 | 1118482 |
| 16 | 524288 | 786432 | 1048576 | 1572864 | 2097152 |

## Висновки

Підписи на основі геш-функцій є перспективним класом постквантових асиметричних криптоалгоритмів. Важливим компонентом криптосистем сімейства SPHINCS є одноразові підписи, зокрема підпис Вінтерніца.

В рамках роботи були проаналізовані загальновідомі одноразові ЕЦП – Лампорта, Лампорта – Діффі, Вінтерніца – та ряд можливих модифікацій, які за певних умов дозволяють досягнути кращих результатів. Особливо цікавим є розширений підпис Лампорта, що зберігає обчислювальну складність та розміри ключів оригінального алгоритму і при цьому дозволяє вдвічі зменшити розмір підпису. Проте, він не має важливої переваги підпису Вінтерніца – можливості повного обчислення публічного ключа з підпису, що є особливо важливим при використанні алгоритму у багаторівневих структуруах гіпердерев, зокрема в підписах сімейства SPHINCS.

**Список літератури:**


1. [Електронний ресурс] https://csrc.nist.gov/projects/post-quantum-cryptography.
2. Jean-Phillippe Aumasson and Guillaume Endignoux.: Gravity- SPHINCS – Submission to the NIST's post-quantum cryptography standardization process. (2017).
3. Daniel J. Bernstein et al.: SPHINCS+ – Submission to the NIST's post-quantum cryptography standardization process. (2019).
4. [Електронний ресурс] https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions.
5. Daniel J. Bernstein et al. Sphincs: practical stateless hash-based signatures. Cryptology ePrint Archive, Report 2014/795, 2014.
6. Leslie Lamport. Constructing digital signatures from a one-way func- tion. Technical. Report SRI-CSL-98, SRI International Computer Science Laboratory, 1979.
7. Ralph C. Merkle. A certified digital signature. In Gilles Brassard, editor, CRYPTO, volume 435 of LNCS, pages 218–238. Springer, 1989.
8. Johannes Buchmann, Erik Dahmen, Sarah Ereth, Andreas Hulsing, and Markus Ruckert. On the security of the Winternitz one-time signature scheme. In A. Nitaj and D. Pointcheval, editors, Africacrypt 2011, volume 6737 of Lecture Notes in Computer Science, pages 363–378. Springer Berlin / Heidelberg, 2011.
9. Andreas Huülsing. W-OTS+ – shorter signatures for hash-based signa- ture schemes. In Amr Youssef, Abderrahmane Nitaj, and Aboul-Ella Hassanien, editors, Progress in Cryptology // AFRICACRYPT 2013, volume 7918 of LNCS, pages 173–188. Springer, 2013.
10. Leonid Reyzin and Natan Reyzin. Better than biba: Short one-time signatures with fast signing and verifying. In Lynn Batten and Jennifer Seberry, editors, Information Security and Privacy, volume 2384 of Lecture Notes in Computer Science, pages 1–47. Springer Berlin / Heidelberg, 2002.
11. Andreas Huülsing, Lea Rausch, and Johannes Buchmann. Optimal parameters for XMSSMT. In Alfredo Cuzzocrea, Christian Kittl, Dimitris E. Simos, Edgar Weippl, and Lida Xu, editors, Security Engineering and Intelligence Informatics, volume 8128 of Lecture Notes in Computer Science, pages 194–208. Springer Berlin Heidelberg, 2013.
12. Ю.І. Горбенко, Т.В. Мельник, І.Д. Горбенко. Аналіз потенційних постквантових механізмів електронних підписів на основі геш-функцій // Радиотехника. 2017. Вып. 189.
13. M.A. Kudinov, E.O. Kiktenko, A.K. Fedorov. Security analysis of the W-OTS+ signature scheme: Updating security bounds – 2020.


*Надійшла до редколегії 30.10.2020*

*Відомості про авторів:*


**Семенець Валерій Васильович** – д-р техн. наук, професор, ректор, Харківський національний університет радіоелектроніки, Україна, e-mail: valery.semenets@nure.ua, ORCID: https://orcid.org/0000-0001-8969-2143

**Марухненко Олександр Сергійович –** студент кафедри безпеки інформаційних технологій, Харківський національний університет радіоелектроніки, Україна, e-mail: oleksandr.marukhnenko@nure.ua, ORCID: https://orcid.org/0000-0002-0583-3752

**Горбенко Іван Дмитрович** – д-р техн. наук, професор, професор кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, Харківський національний університет імені В.Н. Каразіна; головний конструктор АТ «Інститут інформаційних технологій», Україна, e-mail: GorbenkoI@iit.kharkov.ua, ORCID: https://orcid.org/0000-0003-4616-3449

**Халімов Геннадій Зайдулович –** доктор технічних наук, професор, завідувач кафедри безпеки інформаційних технологій, Харківський національний університет радіоелектроніки, Україна, e-mail: hennadii.khalimov@nure.ua, ORCID: https://orcid.org/0000-0002-2054-9186