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

Реферат на тему
Криптографічні примітиви

Роботу виконав:
Юрчук Олексій, ФІ-52мн

3 грудня 2025 р.
м. Київ

ЗМІСТ

Розділ 1

Рівні стійкості криптографічних примітивів

Сучасна криптографія базується на формальних визначеннях безпеки, які дозволяють математично доводити стійкість криптографічних схем [1]. Ці визначення формулюються у вигляді *ігор безпеки* (security games) між супротивником (adversary) та челенджером (challenger), де супротивник намагається порушити якусь властивість криптосистеми [2]. В першому розділі розглянемо основні рівні стійкості криптографічних примітивів: односторонність (one-wayness), нерозрізненість (indistinguishability), семантична стійкість (semantic security) та стійкість до перетворень (non-malleability). Ці поняття аналізуються в контексті різних моделей атак, зокрема атак на основі обраного відкритого тексту (CPA), неадаптивних атак на основі обраного шифротексту (CCA-1) та адаптивних атак на основі обраного шифротексту (CCA-2) [3].

1.1 Моделі атак

Перед переходом безпосередньо до рівнів стійкості необхідно визначити моделі атак, які характеризують спектр можливостей супротивника. Нехай $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ – асиметрична схема шифрування (Public Key Encryption) з простором повідомлень $\mathcal{M}$ та простором шифротекстів $\mathcal{C}$ [1].

1.1.1 Chosen Plaintext Attack (CPA/CMA)

В моделі атаки на основі обраного відкритого тексту супротивник має доступ до відкритого ключа pk і може обчислювати шифротексти для довільних повідомлень за власним вибором. Формально, супротивник $\mathcal{A}$ має оракульний доступ до функції шифрування $\mathsf{Enc}_{\mathsf{pk}}(\cdot)$ (тобто має можливість надсилати запити до функції/алгоритму оракула і отримувати коректні відповіді без знання внутрішнього ключа або його механізму роботи) [4].

Означення 1.1.1 (CPA-супротивник [5])**.**
CPA-супротивником називається ймовірнісний поліноміальний алгоритм $\mathcal{A}$, який отримує на вхід відкритий ключ pk та має доступ до оракула шифрування $\mathsf{Enc}_{\mathsf{pk}}(\cdot)$.

Для детермінованих схем шифрування з відкритим ключем доступ до оракула шифрування не надає додаткової переваги, оскільки супротивник може самостійно обчислити $\mathsf{Enc}_{\mathsf{pk}}(m)$ для будь-якого m [1].

1.1.2 Non-adaptive Chosen Ciphertext attack (CCA-1)

В моделі CCA-1 (також відомій як "lunchtime attack" або Naor-Yung attack), супротивник додатково має доступ до оракула дешифрування $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$, але лише *до* отримання challenge-шифротексту [6].

Означення 1.1.2 (CCA-1 супротивник [5])**.**

CCA-1 супротивником називається ймовірнісний поліноміальний алгоритм $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, де:*

- $\mathcal{A}_1$ *отримує* pk *та має доступ до* $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$*, генерує стан* state*;*
- $\mathcal{A}_2$ *отримує challenge та* state*, але* не має *доступу до* $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$*.*

1.1.3 Adaptive Chosen Ciphertext attack (CCA-2)

Модель CCA-2, запропонована Рекоффом і Саймоном, є "найсильнішою"(найгіршою з точки зору захисту) стандартною моделлю атаки [7]. Супротивник має доступ до оракула дешифрування як до, так і після отримання challenge-шифротексту, з єдиним обмеженням – він не може запитувати дешифрування самого challenge-шифротексту.

Означення 1.1.3 (CCA-2 супротивник [5])**.**

CCA-2 супротивником називається ймовірнісний поліноміальний алгоритм $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, де обидві фази мають доступ до* $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$*, з обмеженням, що* $\mathcal{A}_2$ *не може запитувати дешифрування challenge-шифротексту* c^**.*

Згрупуємо ці атаки у порівняльну таблицю 1.1.

Привілегії	**CPA**	**CCA-1**	**CCA-2**
Доступ до pk	Так	Так	Так
Оракул $\mathsf{Enc}_{\mathsf{pk}}(\cdot)$	Так	Так	Так
Оракул $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$ до challenge	Ні	Так	Так
Оракул $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$ після challenge	Ні	Ні	Так (крім c^*)

Таблиця 1.1: Порівняння моделей атак за можливостями супротивника

1.2 Односторонність (One-Wayness)

Односторонність є *найслабшим* рівнем стійкості для схем шифрування. Вона вимагає, щоб супротивник не міг повністю відновити відкритий текст із шифротексту [3].

Означення 1.2.1 (OW-CPA стійкість)**.**

Нехай $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *– асиметрична схема шифрування, простір можливих атак:* $\mathsf{CPA} \in \{\mathsf{CPA}, \mathsf{CCA1}, \mathsf{CCA2}\}$*. Схема* PKE *називається OW-CPA стійкою, якщо для будь-якого* PPT*(Probabilistic Polynomial-Time)-супротивника* $\mathcal{A}$ *типу CPA:*

$$\mathsf{Adv}^{\mathsf{OW-CPA}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = \Pr\left[\mathcal{A}(\mathsf{pk}, c^*) = m : \begin{array}{c}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ m \xleftarrow{p} \mathcal{M} \\ c^* \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m)\end{array}\right] \leq \mathsf{negl}(\lambda),$$

де λ – параметр безпеки.

Security game для OW-CPA наведена в алгоритмі 1.

Algorithm 1 Game OW-CPA для асиметричного шифрування

Require: Параметр безпеки 1^λ, супротивник $\mathcal{A}$
Ensure: Біт $b \in \{0,1\}$
1: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
2: $m \xleftarrow{p} \mathcal{M}$
3: $c^* \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m)$
4: $m' \leftarrow \mathcal{A}(\mathsf{pk}, c^*)$
5: **if** $m' = m$ **then**
6: **return** 1 ▷ guess successful
7: **else**
8: **return** 0 ▷ guess failed
9: **end if**

Механізм інкапсуляції ключів (Key Encapsulation Mechanism, KEM) є криптографічним примітивом, що складається з трьох алгоритмів $\mathsf{KEM} = (\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$ [8].

Означення 1.2.2 (OW-CPA стійкість KEM)**.**
$KEM = (\mathsf{KeyGen}, \mathsf{Encaps}, \mathsf{Decaps})$ називається OW-CPA стійким, якщо для будь-якого PPT*-супротивника $\mathcal{A}$:*

$$\mathsf{Adv}^{\mathsf{OW-CPA}}_{\mathsf{KEM},\mathcal{A}}(\lambda) = \Pr\left[\mathcal{A}(\mathsf{pk}, c^*) = K : \begin{smallmatrix}(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KeyGen}(1^\lambda)\\(K,c^*)\leftarrow\mathsf{Encaps}(\mathsf{pk})\end{smallmatrix}\right] \leq \mathsf{negl}(\lambda).$$

1.3 Нерозрізненість (Indistinguishability)

Нерозрізненість є значно сильнішим поняттям безпеки, ніж односторонність. Вона вимагає, щоб супротивник не міг отримати *жодної* інформації про відкритий текст із шифротексту [4].

Означення 1.3.1 (IND-CPA стійкість)**.**
Схема шифрування $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ називається IND-CPA стійкою (Indistinguishability under Chosen Plaintext Attack), $\mathsf{CPA} \in \{\mathsf{CPA}, \mathsf{CCA1}, \mathsf{CCA2}\}$, якщо для будь-якого PPT*-супротивника $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$:*

$$\mathsf{Adv}^{\mathsf{IND-CPA}}_{\mathsf{PKE},\mathcal{A}}(\lambda) = \left|\Pr[b' = b] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda),$$

де "Гра" визначена в алгоритмі 2.

Означення 1.3.2 (IND-CPA стійкість KEM)**.**
KEM називається IND-CPA стійким, якщо для будь-якого PPT*-супротивника $\mathcal{A}$:*

$$\mathsf{Adv}^{\mathsf{IND-CPA}}_{\mathsf{KEM},\mathcal{A}}(\lambda) = \left|\Pr[\mathcal{A}(\mathsf{pk}, c^*, K_b) = b] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda),$$

де $K_0 = K$ – справжній ключ з $(K, c^*) \leftarrow \mathsf{Encaps}(\mathsf{pk})$, а $K_1 \xleftarrow{p} \mathcal{K}$ – випадковий ключ.

Algorithm 2 Game IND-CCA2 для асиметричного шифрування

Require: Параметр безпеки 1^λ, супротивник $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$
Ensure: Біт $b' \in \{0,1\}$
1: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
2: $(m_0, m_1, \mathsf{state}) \leftarrow \mathcal{A}_1^{\mathsf{Dec}_{\mathsf{sk}}(\cdot)}(\mathsf{pk})$ ▷ $|m_0| = |m_1|$
3: $b \xleftarrow{p} \{0,1\}$
4: $c^* \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_b)$
5: $b' \leftarrow \mathcal{A}_2^{\mathsf{Dec}_{\mathsf{sk}}(\cdot)}(c^*, \mathsf{state})$ ▷ $\mathcal{A}_2$ не може запитувати $\mathsf{Dec}_{\mathsf{sk}}(c^*)$
6: **return** b'

1.4 Семантична стійкість (Semantic Security)

Семантична стійкість, введена Голдвассер та Мікалі [4], є симуляційним означенням безпеки. Інтуїтивно: схема є семантично стійкою, якщо будь-яку інформацію про відкритий текст, яку можна ефективно обчислити з шифротексту, можна також ефективно обчислити без шифротексту.

Означення 1.4.1 (SS-CPA стійкість)**.**
Схема шифрування PKE *називається SS-CPA стійкою, якщо для будь-якого* PPT*-супротивника $\mathcal{A}$ існує* PPT*-симулятор $\mathcal{S}$ такий, що для будь-якої функції $f : \mathcal{M} \to \{0,1\}^*$ та розподілу $\mathcal{D}$ на $\mathcal{M}$:*

$$\left|\Pr[\mathcal{A}(\mathsf{pk}, \mathsf{Enc}_{\mathsf{pk}}(m)) = f(m)] - \Pr[\mathcal{S}(\mathsf{pk}, 1^{|m|}) = f(m)]\right| \leq \mathsf{negl}(\lambda),$$

де $m \leftarrow \mathcal{D}$.

Твердження 1.4.1 (Еквівалентність IND та SS [4, 9])**.**
Для моделі Chosen Plaintext Attack (CPA) маємо: IND-CPA $\Leftrightarrow$ SS-CPA.

Цей результат був розширений Белларе та ін. [3] на моделі CCA-1 та CCA-2:

$$\mathsf{IND\text{-}CCA1} \Leftrightarrow \mathsf{SS\text{-}CCA1}, \quad \mathsf{IND\text{-}CCA2} \Leftrightarrow \mathsf{SS\text{-}CCA2}.$$

1.5 Стійкість до перетворень (Non-Malleability)

Стійкість до перетворень (non-malleability) є напрямком захисту від атак, де супротивник намагається створити шифротекст, пов'язаний із challenge-шифротекстом [10].

Означення 1.5.1 (NM-CPA стійкість)**.**
Схема PKE *називається NM-CPA стійкою, якщо для будь-якого* PPT*-супротивника $\mathcal{A}$, для будь-якого відношення R та розподілу $\mathcal{D}$:*

$$\Pr\left[R(m,\mathbf{m}')=1 \wedge c^* \notin \mathbf{c}' : \begin{array}{c}(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KeyGen}(1^\lambda)\\ m\leftarrow\mathcal{D}\\ c^*\leftarrow\mathsf{Enc}_{\mathsf{pk}}(m)\\ \mathbf{c}'\leftarrow\mathcal{A}(\mathsf{pk},c^*)\\ \mathbf{m}'\leftarrow\mathsf{Dec}_{\mathsf{sk}}(\mathbf{c}')\end{array}\right] \approx \Pr\left[R(m,\mathbf{m}')=1 : \mathbf{m}'\xleftarrow{m\leftarrow\mathcal{D}}\mathcal{S}(\mathsf{pk},1^{|m|})\right].$$

В моєму розумінні означення [3], схема є NM-стійкою, якщо маючи шифротекст c^*, супротивник не може створити такий вектор шифротекстів $\mathbf{c}'$, дешифрування яких утворює вектор $\mathbf{m}'$, що є лінійною комбінацією оригінального повідомлення m.

1.6 Порівняльний аналіз означень

Означення	**На що спрямований захист**	**Тип означення**
OW (односторонність)	Повне відновлення повідомлення	Обчислювальне
IND (нерозрізненість)	Будь-яка інформація про повідомлення	Game-based
SS (семантична стійкість)	Будь-яка функція від шифротексту	Simulation-based
NM (стійкість до перетворень)	Створення пов'язаних шифротекстів	Simulation-based

Таблиця 1.2: Властивості різних рівнів стійкості

1.7 Ієрархія та імплікації між рівнями стійкості

Між різними рівнями стійкості існують певні імплікаційні співвідношення, які формують ієрархію стійкості [3, 11].

1.7.1 За типом атаки

Для фіксованого рівня стійкості $X \in \{\mathsf{OW}, \mathsf{IND}, \mathsf{SS}, \mathsf{NM}\}$:

$$X\text{-}\mathsf{CCA2} \Rightarrow X\text{-}\mathsf{CCA1} \Rightarrow X\text{-}\mathsf{CPA}.$$

Ці імплікації є односторонніми (зворотні імплікації не виконуються в загальному випадку) [3].

1.7.2 За рівнем стійкості

Для фіксованого типу атаки $\mathsf{CPA} \in \{\mathsf{CPA}, \mathsf{CCA1}, \mathsf{CCA2}\}$ [3, 11]:

$$\mathsf{NM\text{-}CPA} \Rightarrow \mathsf{IND\text{-}CPA} \Leftrightarrow \mathsf{SS\text{-}CPA} \Rightarrow \mathsf{OW\text{-}CPA}.$$

(!) Важливим фактом є те, що для CCA-2 атак нерозрізненість та стійкість до перетворень є еквівалентними поняттямм [11]:

$$\mathsf{IND\text{-}CCA2} \Leftrightarrow \mathsf{NM\text{-}CCA2}.$$

А для CPA ця еквівалентність не виконується:

$$\mathsf{NM\text{-}CPA} \Rightarrow \mathsf{IND\text{-}CPA}, \quad \text{але} \quad \mathsf{IND\text{-}CPA} \not\Rightarrow \mathsf{NM\text{-}CPA}.$$

1.7.3 Загальна ієрархія

Ієрархію рівнів стійкості для асиметричного шифрування можна гарно відобразити рисунком 1.1.

Стрілками позначимо імплікації. IND-CCA2 $\Leftrightarrow$ NM-CCA2 – єдина еквівалентність між IND та NM.

Найвищим рівнем стійкості для схем асиметричного шифрування є IND-CCA2 (еквівалентно NM-CCA2). Цей рівень є "золотим стандартом" для практичних криптосистем [11].

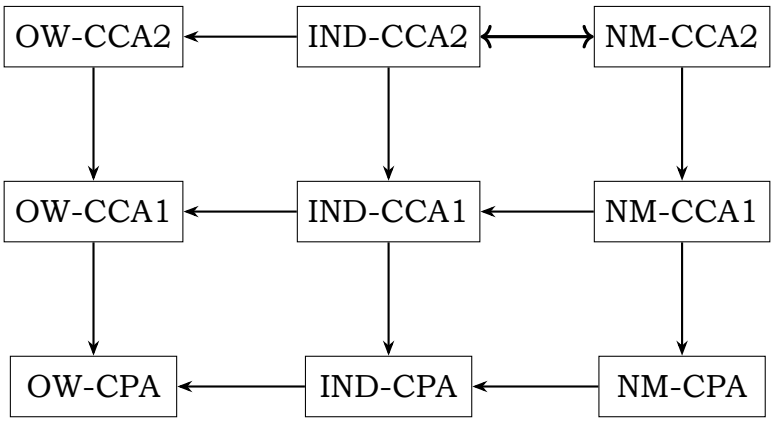

Рис. 1.1: Ієрархія рівнів стійкості криптопримітивів.

1.8 Приклади криптопримітивів

1.8.1 Криптопримітиви з доведеною стійкістю

RSA-OAEP (IND-CCA2)

RSA-OAEP (Optimal Asymmetric Encryption Padding) є стандартизованою схемою шифрування з відкритим ключем [12]. Схема використовує RSA-функцію з використанням оптимального падінгу, що базується на двох різних геш-функціях. Алгоритм шифрування RSA-OAEP є наступним:

Algorithm 3 RSA-OAEP шифрування

Require: Повідомлення m, відкритий ключ (n, e), геш-функції G, H
Ensure: Шифротекст c

1: $r \xleftarrow{p} \{0,1\}^{k_0}$ ▷ Випадкове значення
2: $s \leftarrow (m \| 0^{k_1}) \oplus G(r)$
3: $t \leftarrow r \oplus H(s)$
4: $w \leftarrow s \| t$
5: $c \leftarrow w^e \pmod n$
6: **return** c

Твердження 1.8.1 (Стійкість RSA-OAEP [13])**.**
RSA-OAEP є IND-CCA2 стійкою в моделі випадкового оракула за припущення складності RSA-задачі.

Cramer-Shoup (IND-CCA2 без ROM)

Схема Крамера-Шоупа є першою практичною схемою шифрування з відкритим ключем, для якої доведена IND-CCA2 стійкість у стандартній моделі (без випадкового оракула) [14].

Твердження 1.8.2 (Стійкість Cramer–Shoup [15])**.**
Схема Cramer–Shoup є IND-CCA2 стійкою за припущення DDH (Decisional Diffie–Hellman assumption).

ML-KEM (a.k.a Kyber)

ML-KEM (Module-Lattice-based Key Encapsulation Mechanism), раніше відомий як CRYSTALS-Kyber, є стандартизованим постквантовим KEM [16]. Він був обраний NIST (National Institute of Standards and Technology) як стандарт для постквантової криптографії.

Твердження 1.8.3 (Стійкість ML-KEM [17])**.**
ML-KEM є IND-CCA2 стійким за припущення складності задачі MLWE (Module Learning with Errors).

1.8.2 Криптопримітиви, що не задовольняють певним рівням стійкості

Textbook RSA

"Підручникова" схема RSA (Rivest-Shamir-Adleman without padding) не задовольняє навіть найслабшому рівню нерозрізненості IND-CPA [1].

Доведення:
Нехай $(\mathsf{pk}, \mathsf{sk}) = ((n, e), d)$ – ключова пара RSA. Розглянемо супротивника $\mathcal{A}$, який:

1. Вибирає повідомлення m_0, m_1;
2. Отримує challenge-шифротекст $c^* = m_b^e \pmod n$;
3. Обчислює $c_0 = m_0^e \pmod n$;
4. Якщо $c^* = c_0$, виводить $b' = 0$, інакше $b' = 1$.

Оскільки RSA є детермінованим алгоритмом, то зловмисник $\mathcal{A}$ вгадує правильно з ймовірністю $\mathsf{Adv}^{\mathsf{IND-CPA}}_{\mathsf{RSA},\mathcal{A}} = 1/2$. Окрім цього, Textbook RSA має властивість *мультиплікативності*, що робить її вразливою до атак на перетворення (NM-CPA) [18, 19].

$$\mathsf{Enc}(m_1) \cdot \mathsf{Enc}(m_2) = m_1^e \cdot m_2^e = (m_1 \cdot m_2)^e = \mathsf{Enc}(m_1 \cdot m_2) \pmod n,$$

□

ElGamal

Схема ElGamal є прикладом криптосистеми, яка задовольняє IND-CPA, але не задовольняє IND-CCA1 [20].

Твердження 1.8.4 (Стійкість ElGamal [21])**.**
Схема ElGamal є IND-CPA стійкою за припущення DDH, але не є IND-CCA2 стійкою (і, як наслідок, не є IND-CCA1 стійкою).

Доведення: Нехай $\mathsf{pk} = (G, g, h = g^x)$. Супротивник $\mathcal{A}$ діє наступним чином:

1. Вибирає два повідомлення $m_0, m_1 \in G$;
2. Отримує challenge-шифротекст $c^* = (c_1, c_2) = (g^r, m_b \cdot h^r)$;
3. Формує модифікований шифротекст $c' = (c_1, c_2 \cdot g) = (g^r, m_b \cdot h^r \cdot g)$;
4. Запитує $\mathsf{Dec}(c')$ у фазі після отримання c^* (це дозволено в CCA-2, оскільки $c' \neq c^*$);
5. Отримує $m' = m_b \cdot g$ та обчислює $m_b = m' \cdot g^{-1}$;
6. Виводить $b' = 0$, якщо $m_b = m_0$, інакше $b' = 1$.

Супротивник вгадує правильно з ймовірністю $\mathsf{Adv}^{\mathsf{IND-CCA2}} = 1/2$. □

Ця вразливість пов'язана з malleability. Якщо $(c_1, c_2) = (g^r, m \cdot h^r)$ є шифротекстом для m, то для будь-якого відомого $\delta \in G$:

$$(c_1, c_2 \cdot \delta) = (g^r, m \cdot \delta \cdot h^r) = \mathsf{Enc}(m \cdot \delta),$$

тобто можна отримати валідний шифротекст для $m \cdot \delta$ без знання m. Ця властивість є наслідком мультиплікативності алгоритму ElGamal:

$$\mathsf{Enc}(m_1) \cdot \mathsf{Enc}(m_2) = (g^{r_1+r_2}, m_1 \cdot m_2 \cdot h^{r_1+r_2}) = \mathsf{Enc}(m_1 \cdot m_2).$$

1.8.3 Порівняльна таблиця перелічених алгоритмів

Крипто-примітив	**OW-CPA**	**IND-CPA**	**IND-CCA1**	**IND-CCA2**	**NM-CPA**	**NM-CCA2**
RSA-OAEP	Так	Так	Так	Так***	Так	Так***
Cramer–Shoup	Так	Так	Так	Так**	Так	Так**
ML-KEM (Kyber)	Так	Так	Так	Так*	Так	Так*
Textbook RSA	Так*	Ні	Ні	Ні	Ні	Ні
ElGamal	Так	Так**	Ні†	Ні	Ні	Ні

* — за припущення складності RSA-задачі; ** — за припущення DDH; *** — у моделі випадкового оракула; $\star$ – за припущення MLWE.
† — для ElGamal доведено нестійкість до CCA-2; нестійкість до CCA-1 не має явної простої атаки, але й доказу стійкості немає.

Таблиця 1.3: Порівняння рівнів стійкості криптопримітивів

Можна підбити коротенький підсумок:

1. Ієрархія рівнів стійкості: IND-CCA2 (еквівалентно NM-CCA2) є найвищим рівнем стійкості для схем асиметричного шифрування та механізмів інкапсуляції ключів.
2. Нерозрізненість та семантична стійкість еквівалентні для всіх розглянутих моделей атак (CPA, CCA-1, CCA-2).
3. Для CCA-2 атак IND та NM еквівалентні, але для CPA атак NM є строго сильнішою вимогою.
4. Сучасні криптосистеми (RSA-OAEP, Cramer–Shoup, ML-KEM) розробляються з метою досягнення IND-CCA2 стійкості так званого "золотого стандарту" безпеки.
5. Приклад Textbook RSA демонструє критичну важливість використання падінгу як такового для досягнення навіть найбазовіших рівнів стійкості.

Розділ 2

Рівні стійкості схем цифрового підпису

Схеми цифрового підпису(ЦП) є фундаментальним криптографічним примітивом, що забезпечує автентичність та цілісність повідомлень [2, 22]. На відміну від схем шифрування, де основною метою є конфіденційність, для схем підпису ключовою властивістю є захист від підробок (unforgeability) – неможливість створення валідного підпису без знання секретного ключа [23].

Формально, схема цифрового підпису складається з трьох алгоритмів [1]:

$\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$:

- $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$ — генерація ключової пари;
- $\mathsf{Sign}_{\mathsf{sk}}(m) \to \sigma$ — створення підпису для повідомлення m;
- $\mathsf{Verify}_{\mathsf{pk}}(m, \sigma) \to \{0, 1\}$ — перевірка підпису.

2.1 Моделі атак на схеми цифрового підпису

Існують багато рівнів стійкості, розглянемо два основних – універсальна непідробність (universal unforgeability, UU) та екзистенційна непідробність (existential unforgeability, EU). Їх доцільно розглядати в контексті різних моделей атак, наприклад: KOA (key-only attack), RMA (random message attack) та CPA (chosen message attack). Моделі атак на схеми ЦП класифікуються за обсягом інформації, яка доступна супротивнику [1, 23].

2.1.1 Атака лише з відкритим ключем (KOA)

В моделі KOA (Key-Only Attack) супротивник має доступ лише до відкритого ключа pk. Це найслабша модель атаки, оскільки супротивник не має жодних прикладів валідних підписів [23].

Означення 2.1.1 (KOA-супротивник)**.**
KOA це PPT*(Probabilistic Polynomial-Time)-алгоритм $\mathcal{A}$, який отримує на вхід лише відкритий ключ користувача* pk *та намагається створити валідний підпис.*

2.1.2 Атака з випадково обраними повідомленнями (RMA)

В моделі RMA (Random Message(Plaintext) Attack), також відомій як KPA (Known Plaintext Attack), супротивник перехоплює набір пар (m_i, σ_i), де повідомлення m_i обрані випадковим чином [23].

Означення 2.1.2 (RMA-супротивник)**.**
RMA-супротивником називається PPT*-алгоритм $\mathcal{A}$, який отримує:*

- *відкритий ключ* pk*;*
- *набір пар* $\{(m_1, \sigma_1), \ldots, (m_q, \sigma_q)\}$*, де* $m_i \xleftarrow{p} \mathcal{M}$ *та* $\sigma_i = \mathsf{Sign}_{\mathsf{sk}}(m_i)$.

2.1.3 Атака на основі вибраного plaintext (CPA)

Модель CPA (Chosen Plaintext Attack) є найсильнішою стандартною моделлю атаки. Супротивник має адаптивний (ґрунтуючись на попередньо отриманих результатах) оракульний доступ до функції підпису $\mathsf{Sign}_{\mathsf{sk}}(\cdot)$ і може запитувати підписи для довільних повідомлень за власним вибором [23].

Означення 2.1.3 (CPA-супротивник)**.**
CPA-супротивником називається такий поліноміальний алгоритм $\mathcal{A}^{\mathsf{Sign}_{\mathsf{sk}}(\cdot)}$*, який:*

- *отримуючи відкритий ключ* pk*;*
- *і маючи адаптивний оракульний доступ до* $\mathsf{Sign}_{\mathsf{sk}}(\cdot)$*;*
- *може робити поліноміальну кількість запитів до оракула для витягання деталей про* sk.

Всі перелічені атаки можна ґарненько звести до таблиці 2.1.

Преференція	**KOA**	**RMA**	**CPA**
Sign examples	Ні	Так (випадкові m_i)	Так (обрані m_i)
Адаптивність запитів	Ні	Ні	Так
Загроза	Найслабша	Середня	Найсильніша

Таблиця 2.1: Порівняння моделей атак на схему цифрового підпису

Ієрархія моделей атак

Між моделями атак існує певна ієрархія залежно від їх сили [23]:

$$\mathsf{CPA} \succ \mathsf{RMA} \succ \mathsf{KOA},$$

де під позначенням $A \succ B$ розуміємо, що модель A надає супротивнику більше можливостей, ніж модель B. І відповідно стійкість до більш сильної атаки (позначимо її X) включає в себе і стійкість до слабшої:

$$X\text{-}\mathsf{CPA} \Rightarrow X\text{-}\mathsf{RMA} \Rightarrow X\text{-}\mathsf{KOA}$$

2.2 Рівні непідробності

Рівні непідробності розрізняють залежно від того, що саме вважається успішною підробкою. Розглянемо три основні види.

2.2.1 Універсальна непідробність (Universal Unforgeability)

Універсальна непідробність (UU) вимагає, щоб супротивник не міг підробити підпис для *заданого* повідомлення m^*, яке обирається челенджером [23].

Означення 2.2.1 (UU-ATK стійкість)**.**

Схема підпису $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ *називається UU-ATK стійкою, де* $\mathsf{ATK} \in \{\mathsf{KOA}, \mathsf{RMA}, \mathsf{CPA}\}$*, якщо для будь-якого* PPT*-супротивника* $\mathcal{A}$*:*

$$\mathsf{Adv}^{\mathsf{UU-ATK}}_{\Sigma,\mathcal{A}}(\lambda) = \Pr\left[\mathsf{Verify}_{\mathsf{pk}}(m^*, \sigma^*) = 1 : \begin{smallmatrix}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ m^* \xleftarrow{p} \mathcal{M} \\ \sigma^* \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pk}, m^*)\end{smallmatrix}\right] \leq \mathsf{negl}(\lambda),$$

де $\mathcal{O}$ *– оракул, визначений моделлю ATK.*

P.S. В моделі CPA супротивник не може запитувати $\mathsf{Sign}_{\mathsf{sk}}(m^*)$*.*

Алгоритм ”гри” UU-CPA наведемо в алгоритмі 4.

Algorithm 4 Game UU-CPA для схеми цифрового підпису

Require: Параметр безпеки 1^λ
Ensure: Біт $b \in \{0, 1\}$
1: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
2: $m^* \xleftarrow{p} \mathcal{M}$ ▷ Челенджер обирає цільове повідомлення
3: $Q \leftarrow \emptyset$ ▷ Множина запитаних повідомлень
4: $\sigma^* \leftarrow \mathcal{A}^{\mathsf{Sign}_{\mathsf{sk}}(\cdot)}(\mathsf{pk}, m^*)$ ▷ $\mathcal{A}$ не може запитувати $\mathsf{Sign}_{\mathsf{sk}}(m^*)$
5: **if** $\mathsf{Verify}_{\mathsf{pk}}(m^*, \sigma^*) = 1$ **and** $m^* \notin Q$ **then**
6: **return** 1 ▷ Успішна підробка підпису
7: **else**
8: **return** 0
9: **end if**

2.2.2 Екзистенційна непідробність (Existential Unforgeability)

Екзистенційна несфальсифікованість (EU) є більш сильним поняттям. Вона вимагає, щоб супротивник не міг підробити підпис для *будь-якого* повідомлення, яке підписант раніше не підписував [24]. Це є ”золотим стандартом” безпеки для цифрових підписів.

Означення 2.2.2 (EU-ATK стійкість)**.**

Схема підпису Σ *є EU-ATK стійкою, якщо для будь-якого* PPT*-algorithm* $\mathcal{A}$*:*

$$\mathsf{Adv}^{\mathsf{EU-ATK}}_{\Sigma,\mathcal{A}}(\lambda) = \Pr\left[\begin{smallmatrix}\mathsf{Verify}_{\mathsf{pk}}(m^*,\sigma^*)=1 \\ \wedge\, m^* \notin Q\end{smallmatrix} : \begin{smallmatrix}(\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ (m^*,\sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{pk})\end{smallmatrix}\right] \leq \mathsf{negl}(\lambda),$$

де Q *– множина повідомлень, для яких атакуючий* $\mathcal{A}$ *отримав підписи.*

EU-CPA також доволі часто позначають в літературі як EUF-CMA (Existential Unforgeability under Chosen Message Attack) [1].

Algorithm 5 Game EU-CPA (EUF-CPA) для схеми цифрового підпису

Require: Параметр безпеки 1^λ, супротивник $\mathcal{A}$
Ensure: Біт $b \in \{0, 1\}$
1: $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$
2: $Q \leftarrow \emptyset$
3: $(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathsf{Sign}_{\mathsf{sk}}(\cdot)}(\mathsf{pk})$ ▷ $\mathcal{A}$ сам обирає m^*
4: **if** $\mathsf{Verify}_{\mathsf{pk}}(m^*, \sigma^*) = 1$ **and** $m^* \notin Q$ **then**
5: **return** 1 ▷ Успішна підробка
6: **else**
7: **return** 0
8: **end if**

2.2.3 Сильна екзистенційна непідробність (sEU)

Виділяють також ще одне поняття – *сильна екзистенційна непідробність* (Strong Unforgeability under Chosen Message Attack, sEU a.k.a. SUF), яке додатково запобігає фальсифікації зловмисником дійсного підпису для будь-якого нового повідомлення, включаючи те, яке вже було підписано законним підписувачем [25].

Означення 2.2.3 (sEU-CPA стійкість)**.**
Схема Σ є sEU-CPA стійкою, якщо супротивник не може створити таку пару (m^*, σ^*), що $\mathsf{Verify}_{\mathsf{pk}}(m^*, \sigma^*) = 1 \wedge (m^*, \sigma^*) \notin Q$, де Q – множина всіх пар (повідомлення, підпис), отриманих від оракула.

Зв'язок між рівнями непідробності

Для фіксованого типу атаки ATK [1, 23]:

$$\mathsf{sEU\text{-}ATK} \Rightarrow \mathsf{EU\text{-}ATK} \Rightarrow \mathsf{UU\text{-}ATK}.$$

Ці імплікації є строгими – зворотні імплікації не виконуються в загальному випадку.

2.3 Важливість sEU-CPA (на практиці)

На перший погляд, різниця між EU-CMA та sEU-CMA може здаватися суто теоретичною: навіщо забороняти створення іншого підпису для вже підписаного повідомлення? У рандомізованих схемах підпису, де для одного повідомлення m може існувати багато валідних підписів ця відмінність є критичною [25].

Розглянемо до прикладу схему Шнора (більш детально про неї в 2.5.2). Там підпис має вигляд $\sigma = (e, s)$, де $e = H(g^k \| m)$ залежить від випадкового k. Для одного повідомлення m з різними значеннями k отримаємо різні валідні підписи $\sigma_1, \sigma_2, \ldots$.
Нехай $\mathcal{A}$ запитав $\mathsf{Sign}(m)$ і отримав $\sigma_1 = (e_1, s_1)$:

- **EU-CMA**: супротивник повинен підробити підпис для *нового* довільно обраного повідомлення $m^* \neq m$. Оскільки створення іншого валідного підпису σ_2 для того ж m не є порушенням.
- **sEU-CMA**: супротивник не може створити *жодну* нову пару (m^*, σ^*), таку яку не було раніше видано оракулом як відповідь на якийсь запит, навіть (m, σ_2), де $\sigma_2 \neq \sigma_1$.

Дана sEU-CPA стійкість важлива в тих протоколах, де підпис використовується як унікальний ідентифікатор певної транзакції або де дублювання підпису може призвести до replay-атак [25].

2.4 Ієрархія рівнів стійкості

Найвищим стандартним рівнем стійкості для схем цифрового підпису вважають sEU-CPA (SUF-CPA), а практичним "золотим стандартом" є EU-CPA (EUF-CPA) [1].

Повна ієрархія рівнів стійкості для схем цифрового підпису зображена на рисунку 2.1. Стрілками позначимо імплікації (від сильнішого до слабшого).

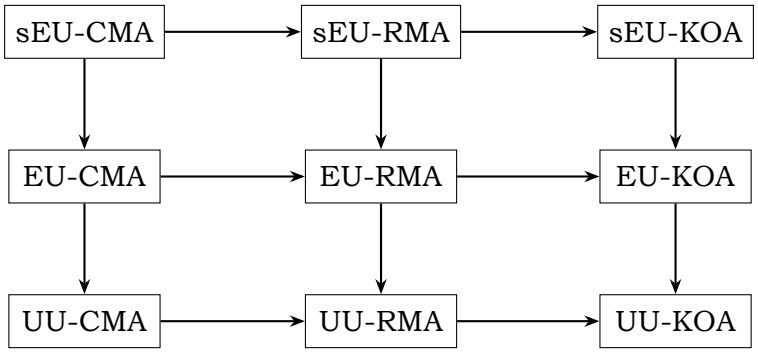

Рис. 2.1: Ієрархія рівнів стійкості схем цифрового підпису.

2.5 Приклади криптопримітивів

2.5.1 Схеми з доведеною EU-CPA стійкістю

RSA-PSS (EU-CPA)

RSA-PSS (Probabilistic Signature Scheme) є стандартизованою схемою підпису, розробленою Белларе та Рогавеєм [26]. На відміну від детермінованого RSA-підпису, PSS використовує рандомізацію. Алгоритм підпису RSA-PSS є наступним:

Algorithm 6 RSA-PSS підпис

Require: Повідомлення m, секретний ключ d, модуль n, геш-функції H, G
Ensure: Підпис σ

1: $r \xleftarrow{p} \{0,1\}^{k_0}$ ▷ Випадкове значення рандомізації
2: $w \leftarrow H(m \| r)$
3: $r^* \leftarrow G(w) \oplus r$
4: $y \leftarrow 0 \| w \| r^*$
5: $\sigma \leftarrow y^d \pmod n$
6: **return** σ

Твердження 2.5.1 (Стійкість RSA-PSS [26])**.**

RSA-PSS є EU-CPA стійкою в моделі випадкового оракула (Random Oracle Model (ROM)) за припущення складності RSA-задачі.

Schnorr (EU-CPA)

Схема Шнорра є схемою підпису на основі розв'язанні задачі дискретного логарифма [27]. Вона є основою для багатьох сучасних схем підпису, наприклад, EdDSA.

Твердження 2.5.2 (Стійкість Schnorr [28])**.**
Схема Schnorr є EU-CPA стійкою в моделі випадкового оракула за припущення складності задачі дискретного логарифма (DL).

Схема виглядає наступним чином:

Algorithm 7 Схема підпису Schnorr

Генерація ключів $\mathsf{KeyGen}(1^\lambda)$:
1: $x \xleftarrow{p} \mathbb{Z}_q$
2: **return** $(\mathsf{pk} = g^x,\ \mathsf{sk} = x)$

Підпис $\mathsf{Sign}_{\mathsf{sk}}(m)$:
3: $k \xleftarrow{p} \mathbb{Z}_q$
4: $r \leftarrow g^k$
5: $e \leftarrow H(r \| m)$
6: $s \leftarrow k + x \cdot e \pmod q$
7: **return** $\sigma = (e, s)$

Верифікація $\mathsf{Verify}_{\mathsf{pk}}(m, \sigma = (e, s))$:
8: $r' \leftarrow g^s \cdot \mathsf{pk}^{-e}$
9: **return** $(H(r' \| m) \stackrel{?}{=} e)$

ECDSA (EU-CPA)

Згадана мною раніше ECDSA (Elliptic Curve Digital Signature Algorithm) є широко використовуваною схемою підпису, стандартизованою у вже застарілому FIPS 186-4 [29]. (Буквально на початку 2024 року вийшло оновлення)

Твердження 2.5.3 (Стійкість ECDSA [30])**.**
ECDSA є EU-CPA стійкою в моделі generic group за припущення складності ECDLP (Elliptic Curve Discrete Logarithm Problem).

ML-DSA (Dilithium) – постквантовий підпис

ML-DSA (Module-Lattice-based Digital Signature Algorithm), раніше відомий як CRYSTALS-Dilithium, є стандартизованим NIST постквантовим підписом (той самий новий, 24 року стандарт) [31].

Він є EU-CPA стійким у припущені складності задач MLWE (learning with errors) та MSIS (short integer solution).

2.5.2 Схеми без певних рівнів стійкості

Textbook RSA signature

"Підручникова" схема RSA-підпису, де $\sigma = m^d \pmod n$, є вразливою навіть до найслабшої атаки [1]. Тобто textbook RSA підпис не є EU-KOA стійким, оскільки супротивник, маючи лише $\mathsf{pk} = (n, e)$, може здійснити екзистенційну підробку:

1. Вибрає довільне $\sigma^* \in \mathbb{Z}_n^*$;
2. Обчислює $m^* = (\sigma^*)^e \pmod n$;
3. Пара (m^*, σ^*) є валідним підписом, оскільки за побудовою $(\sigma^*)^e \equiv m^* \pmod n$, тому $\mathsf{Verify}_{\mathsf{pk}}(m^*, \sigma^*) = 1$ – підпис дійсний.

Ця атака демонструє проблему з інверсією: супротивник *спочатку* обирає підпис, а потім обчислює відповідне повідомлення. Оскільки значення m^* повністю визначається вибором σ^*, то m^* як читабельний людиною текст не матиме практичного сенсу, але це все одно є порушенням екзистенційної непідробності, бо вимагається захист від підробки *будь-якого нового* повідомлення.

Textbook RSA також має властивість *мультиплікативності*, тому крім інверсійної атаки, можлива підробка за наявності підписів інших повідомлень – в моделях RMA (Random Message Attack) та CMA (Chosen Message Attack) [1]:

Твердження 2.5.4 (Мультиплікативна атака на RSA підпис)**.**
Якщо супротивник має підписи $\sigma_1 = m_1^d$ та $\sigma_2 = m_2^d$ для повідомлень m_1, m_2, він може обчислити валідний підпис для $m^* = m_1 \cdot m_2 \pmod n$:

$$\sigma^* = \sigma_1 \cdot \sigma_2 = m_1^d \cdot m_2^d = (m_1 \cdot m_2)^d \pmod n = (m^*)^d.$$

Це показує, що Textbook RSA не є EU-R/CMA стійкою.

ElGamal signature

Оригінальна схема підпису ElGamal [20] має кілька відомих вразливостей. Наприклад, вона є вразливою до екзистенційної підробки в моделі KOA (див. 2.1.1), якщо не використовувати геш-функцію.

Доведення:
Нехай p – велике просте число, g – генератор мультиплікативної групи $\mathbb{Z}_p^*$, обрано секретний ключ $\mathsf{sk} = x \xleftarrow{p} \mathbb{Z}_{p-1}$ та обраховано відкритий ключ $\mathsf{pk} = (p, g, y = g^x \pmod p)$. Підпис для повідомлення m має вигляд (r, s), де $r = g^k \pmod p$, k – випадкове значення, та виконується рівняння верифікації: $g^m = y^r \cdot r^s \pmod p$.

Фальсифікація може відбутися наступним чином:

1. Обирається $a, b \in \mathbb{Z}_{p-1}^*$, $\gcd(b, p-1) = 1$;
2. Обчислюється $r = g^a \cdot y^b \pmod p$;
3. Обчислюється $s = -r \cdot b^{-1} \pmod{p-1}$;
4. Обчислюється $m^* = a \cdot s \pmod{p-1}$, $m^* = m$.

Пара (r, s) буде валідним підписом для m. □

Використання криптографічної геш-функції H (тобто підписуємо $H(m)$ замість вихідного повідомлення m) запобігає цій атаці.

2.5.3 Порівняльна таблиця розглянутих алгоритмів підпису

Схема підпису	**UU-KOA**	**UU-CPA**	**EU-KOA**	**EU-RMA**	**EU-CPA**	**sEU-CPA**
RSA-PSS	Так	Так	Так	Так	Так*	Так*
Schnorr	Так	Так	Так	Так	Так*	Ні**
ECDSA	Так	Так	Так	Так	Так***	Ні**
ML-DSA	Так	Так	Так	Так	Так*	Так*
Textbook RSA ElGamal (без H)	Ні	Ні	Ні	Ні	Ні	Ні

* — в моделі випадкового оракула (ROM); ** — детерміновані (не ймовірнісні) версії; *** — за припущення складності ECDLP; * — за припущення Module-LWE/SIS.

Таблиця 2.2: Порівняння рівнів стійкості схем цифрового підпису

Проміжний підсумок розділу:

1. Моделі атак характеризуються залежно від рівня загрози (KOA $\prec$ RMA $\prec$ CPA) і стійкість до сильнішої включає в себе стійкість до слабшої;
2. Рівні непідробності залежать від обмежень (UU $\prec$ EU $\prec$ sEU);
3. Практичним стандартом безпеки для схем підпису є EU-CPA (EUF-CPA). Сучасні схеми, такі як RSA-PSS, Schnorr, ECDSA, ML-DSA, відповідають йому;
4. Геш-функції є важливими, оскільки їх використання є критичним для досягнення EU-CPA стійкості (в тому ж ElGamal).

Розділ 3

Аналіз стійкості криптографічного примітиву на прикладі TiGER

TiGER (Tiny-polynomial based Ring-LWE/LWR scheme) є постквантовим механізмом інкапсуляції ключів (KEM), розробленим у рамках конкурсу KpqC (Korean Post-Quantum Cryptography Competition) [32]. Алгоритм базується на математичних задачах RLWE (Ring Learning With Errors) та RLWR (Ring Learning With Rounding), які вважаються стійкими до атак як на класичних, так і на квантових комп'ютерах [33, 34].

3.1 Основні характеристики TiGER. Його "родзинки"

TiGER пропонує три рівні безпеки, що відповідають категоріям NIST [35]:

- **TiGER128:** Рівень безпеки NIST-1 (еквівалент AES-128);
- **TiGER192:** Рівень безпеки NIST-3 (еквівалент AES-192);
- **TiGER256:** Рівень безпеки NIST-5 (еквівалент AES-256).

Ключовими інноваціями TiGER є [32]:

1. Гібридний підхід RLWE/RLWR для балансу безпеки та ефективності;
2. Використання подвійної системи корекції помилок (XEf + D2);
3. Використання розріджених тернарних секретів з ваговою нормою $h_s = 274$;
4. Застосування перетворення Fujisaki-Okamoto з неявним відхиленням (implicit rejection).

3.2 Теоретичні основи безпеки

3.2.1 Математична база: RLWE та RLWR

Означення 3.2.1 (RLWE задача)**.**

Задача RLWE (Ring Learning With Errors), вперше формалізована Lyubashevsky, Peikert та Regev [33].
Нехай $R = \mathbb{Z}[x]/(x^n + 1)$, $R_q = \mathbb{Z}_q[x]/(x^n + 1)$ *– кільце многочленів за модулем циклотомічного многочлена, та* χ *– дискретний гаусівський розподіл ймовірностей на* R_q*. Розпізнавальна задача Decision-RLWE*$_{n,q,\chi}$ *полягає у розрізненні наступних розподілів:*

- $(a, a \cdot s + e)$*, де обрано* $a \in R_q$ *(рівномірний розподіл),* $s \in R_q$ *фіксоване,* $e \xleftarrow{p} \chi$*;*

- (a, u), *де* a, u *обрані рівномірно з* R_q.

Означення 3.2.2 (RLWR задача)**.**

Задача RLWR (Ring Learning With Rounding).
Нехай $R_q = \mathbb{Z}_q[x]/(x^n + 1)$, $R_p = \mathbb{Z}_p[x]/(x^n + 1)$, *де* $p < q$. *Для кожного коефіцієнта многочлена окремо застосовуються детерміністична функція округлення до найближчого цілого [34]:*

$$\lfloor f \rceil_p = \sum_{i=0}^{n-1} \left\lfloor \frac{p}{q} \cdot f_i \right\rceil x^i \bmod p.$$

Розпізнавальна задача Decision-RLWR$_{n,q,p}$ *полягає у розрізненні:*

- $(a, \lfloor a \cdot s \rceil_p)$, *де* $a \in R_q$ *обрано рівномірно,* $s \in R_q$ *фіксоване;*
- (a, u), *де* $a \in R_q$, $u \in R_p$ *обрані рівномірно.*

Варто завважити, що при певних параметрах n, q, p, задача RLWR$_{n,q,p}$ є не легшою за задачу RLWE$_{n,q,\chi}$ з розподілом помилок χ, що відповідає помилці округлення.

3.2.2 Архітектура TiGER KEM

TiGER.KEM будується з застосуванням перетворення Fujisaki-Okamoto [36] для базової PKE схеми:

Algorithm 8 TiGER.KEM.Encaps

Require: Відкритий ключ pk
Ensure: Спільний ключ K, шифротекст ct
1: $m \xleftarrow{p} \{0,1\}^{256}$
2: $r \leftarrow G(m, H(pk))$
3: $ct \leftarrow \text{TiGER.PKE.Enc}(pk, m; r)$
4: $K \leftarrow H(m, ct)$
5: **return** (K, ct)

Algorithm 9 TiGER.KEM.Decaps

Require: Секретний ключ sk, шифротекст ct
Ensure: Спільний ключ K або $\bar{K}$
1: $m' \leftarrow \text{TiGER.PKE.Dec}(sk, ct)$
2: $r' \leftarrow G(m', H(pk))$
3: $ct' \leftarrow \text{TiGER.PKE.Enc}(pk, m'; r')$
4: **if** $ct' = ct$ **then**
5: $\quad K \leftarrow H(m', ct)$
6: **else**
7: $\quad K \leftarrow \bar{K} = H(z, ct)$ ▷ Implicit rejection (неявне відхилення)
8: **end if**
9: **return** K

(!) Використання перетворення FO забезпечує IND-CCA безпеку в моделі квантового випадкового оракула (QROM) [37].

3.3 Аналіз атак на TiGER

3.3.1 Решіткові атаки

Решіткові атаки (a.k.a. Lattice reduction attack) є основним класом атак на RLWE та RLWR, які спрямовані на розв'язання задач SVP (Shortest Vector Problem) або CVP (Closest Vector Problem) [38]. Найбільш відомим є алгоритм BKZ (Block Korkine-Zolotarev) редукції [39] у поєднанні з sieving методами [40].

Для оцінки безпеки TiGER проти квантових атак використовується модель Core-SVP (Short Vector Problem) [41], що вимірює складність вирішення SVP на решітці розмірності β (block dimension) і порогове значення для здійснення успішної атаки складає $\beta = 483$.

В класичній моделі обчислень найкращими класичними алгоритми (BKZ + sieving) досягається:

$$T_{\text{classical}} \approx 2^{0.292\beta + o(\beta)} = 2^{0.292 \cdot 483 + 16.4} \approx 2^{157} \text{ операцій}$$

В квантовому всесвіті, при застосуванні квантових версії sieving алгоритмів (див. [42]) складність досягає:

$$T_{\text{quantum}} \approx 2^{0.265\beta + o(\beta)} = 2^{0.265 \cdot 483 + 16.4} \approx 2^{144} \text{ операцій}$$

Решіткові атаки є *універсальними* для всіх RLWE/RLWR схем, включно з TiGER. Проте, параметри TiGER обрані доволі консервативно для забезпечення достатньої безпеки:

Параметр	**Значення**
Розмірність решітки d	1024
Core-SVP β	483
Класична складність	$\approx 2^{157}$ операцій
Квантова складність	$\approx 2^{144}$ операцій
Класична безпека	≈ 143 біт
Квантова безпека (NIST 1)	128 біт

Таблиця 3.1: Складність решіткової атаки на TiGER128

Навіть з урахуванням можливих майбутніх покращень квантових алгоритмів можна бачити, що злам TiGER128 вимагає 2^{128} операцій, що є практично нереалістичним.

Статус: Атака застосовна, але неефективна

Отже як висновок: решіткові атаки не становлять практичної загрози для TiGER при заданих параметрах.

3.3.2 Meet-in-the-Middle атака на RLWE

Meet-LWE атака, розроблена німецьким криптографом Александром Меем (Alexander May) [43], експлуатує розрідженість секретних ключів. Ідея полягає у розбитті секрету на дві частини та пошуку кожної частини окремо з подальшим їх поєднанням (meet-in-the-middle стратегія). Для секрету **s** з ваговою нормою h_s, складність становить:

$$T_{\text{MitM}} \approx 2^{c \cdot h_s \log_2(3)},$$

де $c < 1$ – деяка константа, що залежить від параметрів атаки [43].

Для TiGER $h_s = 274$, і Meet-LWE атака досягає квантової складності:

$$T_{\text{quantum}} \approx 2^{0.25 \cdot 274 \cdot \log_2(3)} \approx 2^{109.5}.$$

Це нижче цільового рівня 128 біт, але:

- Атака вимагає експоненційної пам'яті: $2^{109.5}$ квантових кубітів для збереження проміжних станів;
- Практична реалізація такої атаки на квантовому комп'ютері є нереалістичною навіть у віддаленому майбутньому;
- Для TiGER192 і TiGER256 складність зростає до 2^{164} та 2^{219} відповідно, що значно перевищує можливості для атаки.

Статус: Частково застосовна

При порівнянні з решітковими атаками, BKZ залишається більш ефективною проти TiGER128 (2^{128} vs $2^{109.5}$), але Meet-LWE є теоретично швидша, але практично нереалізованою через величезні вимоги до об'ємів квантової пам'яті.

3.3.3 Атаки на ймовірність помилок декапсуляції

Атаки типу failure boosting [44, 45] експлуатують ненульову ймовірність помилок декапсуляції (Decapsulation/Decryption Failure Probability/Rate, DFP/DFR). Їх ідея полягає у:

1. Генерації спеціальних шифротекстів, які максимізують DFP;
2. Подача цих шифротекстів до оракула декапсуляції;
3. Спостереження успіху/невдачі декапсуляції;
4. Поступове вивідування інформації про секретний ключ.

DFP – це ймовірність того, що правильно сформований шифротекст декапсулюється некоректно, тобто:

$$\text{DFP} = \Pr[\text{Decaps}(sk, \text{Encaps}(pk)) \neq K] = \delta$$

А кількість необхідних запитів для витягування секрету становить $O(1/\delta)$ [44].
TiGER має два рівні захисту від цієї атаки:

- Перше – це дуже низький DFP (досягається завдяки подвійній корекції помилок (XEf + D2)). З специфікації TiGER [32] (заявлена DFP) складає: $\text{DFP}_{\text{TiGER128}} = 2^{-120}$. Це означає, що атакуючому потрібно 2^{120} запитів для спостереження хоча б однієї помилки.
- Друге – це Implicit Rejection (IR). Перетворення $\text{FO}_m^{\not\perp}$ з неявним відхиленням [37] гарантує нам, що при невдалій перевірці $ct' \neq ct$ алгоритм не повертає помилку $\perp$, а замість цього повертається псевдовипадковий ключ $\bar{K} = H(z, ct)$, і атакуючий не може відрізнити успішну декапсуляцію від невдалої. Схематично це зображено на малюнку 3.1.

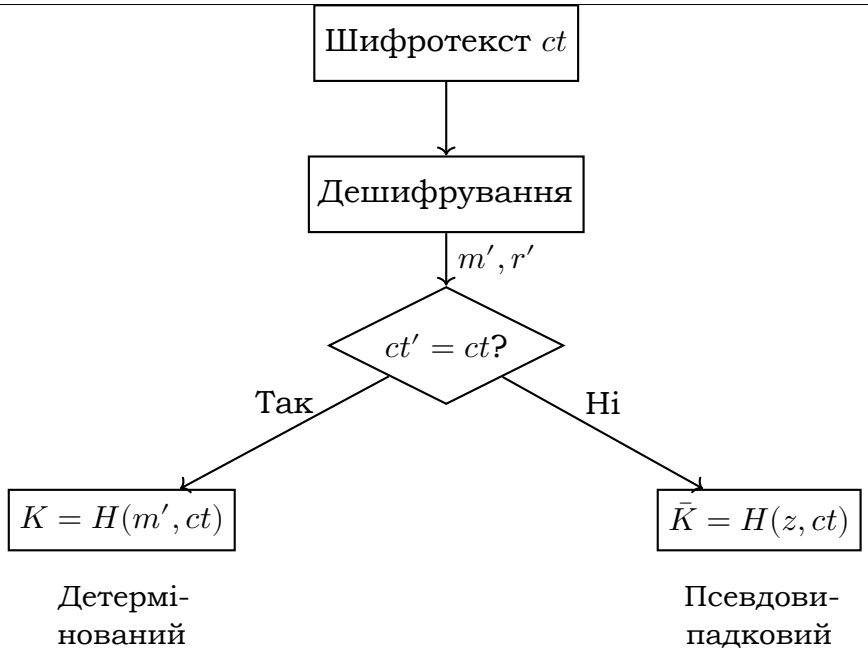

Рис. 3.1: Схема implicit rejection у TiGER

Статус: Не застосовна

Атаки на DFP повністю мітиговані в TiGER завдяки вдалій комбінації низької ймовірності помилок DFP та використанні implicit rejection.

3.3.4 Атаки з вибором шифротексту (CCA)

Атаки з вибраним шифротекстом a.k.a. CCA (Chosen Ciphertext Attack) [19] є одними з найнебезпечніших атак на криптосистеми. В 2024 році von Berg з колегами опублікував роботу [46], що аналізує CCA атаку на PKE схеми на основі решіток, включаючи варіанти наближені до TiGER. Ці атаки дозволяють атакуючому:

- Подавати довільні (в т.ч. модифіковані) шифротексти на оракул декапсуляції;
- Отримувати результати декапсуляції або спостерігати побічні ефекти (помилки);
- Витягати інформацію про секретний ключ або повідомлення.

CCA атаки особливо ефективні проти базових PKE схем без додаткових перетворень [19].

TiGER використовує *не базову PKE*, а KEM з перетворенням Fujisaki-Okamoto (FO), яке перетворює IND-CPA безпечну PKE в IND-CCA2 безпечний KEM. Це перетворення захищає від CCA атак, бо:

1. **Re-encryption check:** Дешифроване μ' використовується для повторного шифрування. Якщо результат не збігається з ct, значить ct був модифікований атакуючим;
2. **Детермінованість:** Шифрування використовує $H'(\mu)$ як seed для помилок, тому одне μ завжди видає той самий ct (для даного pk). Ключ $K = H(\mu)$ повністю визначається повідомленням, тому модифікація ct дає інший μ' та, відповідно, інший K', що виглядає як випадкове значення для атакуючого.
3. **Implicit rejection:** Навіть якщо перевірка провалилася, атакуючий не дізнається про це, оскільки отримує псевдовипадковий $\bar{K}$;

Формальне доведення IND-CCA безпеки в моделі QROM наведено у [37], яке демонструє що $\mathrm{FO}_m^{\not\perp}$ перетворення забезпечує IND-CCA2 безпеку за умов:

- Базова PKE є IND-CPA безпечною;
- Геш-функції H, H' моделюються як випадкові оракули (ROM);
- DFP/R схеми є достатньо малим значенням.

Статус: Не застосовна

Висновок – TiGER є стійким до CCA атак завдяки FO перетворенню з implicit rejection (IR).

3.3.5 Атаки на коди корекції помилок

Атака з урахуванням(спрямована на) XEf коду [47] намагається експлуатувати алгебраїчну структуру кодів корекції помилок. Ідея полягає в маніпуляції шифротекстами для створення специфічних синдромів і спостереженні поведінки декодера (успіх/невдача корекції).

Синдром це лінійна операція XOR над $\mathbb{F}_2$, а це означає, що:

$$s(\mathbf{m}_1 \oplus \mathbf{m}_2) = s(\mathbf{m}_1) \oplus s(\mathbf{m}_2)$$

Припустимо, що зловмисник знає частину бітів повідомлення μ та хоче знайти решту, використовуючи лінійність синдромів. Він зіштовхнеться з наступними проблемами:

1. Синдроми є внутрішньою частиною коду корекції та не передаються разом з шифротекстом;
2. Навіть якщо атакуючому вдасться відтворити процес кодування, йому потрібно знати *точні помилки* від RLWE/компресії (а вони випадкові);
3. Оскільки без знання помилок, система лінійних рівнянь на синдроми має занадто багато невідомих:
   - 32 синдроми (по одному на блок) $\rightarrow$ 32 рівняння;
   - Невідомі: 256 біт μ + помилки від RLWE (кожна має ентропію $\approx n \cdot h_e = 512 \cdot 274 \approx 140,000$ біт) $\rightarrow$ система сильно недовизначена.

Атаки, що спрямовані на XEf код корекції помилок є небезпечними, але TiGER має кілька рівнів захисту, що робить ці атаки практично нездійсненними:

1. **Високий рівень шуму** – RLWE помилки та компресія маскують структуру коду;
2. **Подвійна корекція (XEf + D2)** — навіть якщо XEf зламано, D2 надає додатковий захист;
3. **Implicit rejection** – унеможливлює спостереження success/failure декапсуляції;
4. **FO перетворення** — re-encryption check виявляє модифіковані шифротексти.

Статус: Теоретично застосовна, практично неефективна

Висновок: структура XEf не створює додаткових вразливостей у TiGER завдяки багаторівневому захисту.

3.3.6 Side-channel attacks

Side-channel атаки експлуатують фізичні характеристики при реалізації алгоритму: такі як час виконання, споживання енергії, електромагнітне випромінювання, індукування помилок задля обходу перевірок, аналіз паттернів доступу до кешу, замість безпосередньої атаки на математичну структуру. Для TiGER, як і для всіх постквантових алгоритмів, side-channel атаки становлять серйозну практичну загрозу [48, 49].

Side-channel атаки є *найсерйознішою практичною загрозою* для TiGER, оскільки майже ідеальна математична конструкція не захищає від витоку інформації через фізичні канали.

Тип атаки	**Що відстежується**	**Цільова інформація**
Timing attack	Час виконання операцій	Секретний ключ $\mathbf{s}$, проміжні значення μ
Power analysis (SPA/DPA)	Споживання енергії	Біти секретного ключа $\mathbf{s}$ при множенні
EM analysis	Електромагнітне випромінювання	Аналогічно до power analysis
Cache timing	Паттерни доступу до кешу	Індекси ненульових коефіцієнтів $\mathbf{s}$
Fault injection	Індукування помилок	Обхід перевірок Decaps

Таблиця 3.2: Типи side-channel атак

Контрзаходи, які можливо запровадити для зменшення ризиків side-channel атак:

1. **Constant-time реалізація:** Всі операції виконуються за фіксований час незалежно від розміру вхідних даних [50];
2. **Masking:** Розділення секрету $\mathbf{s}$ на d випадкових часток [51]:

$$\mathbf{s} = \mathbf{s}_0 \oplus \mathbf{s}_1 \oplus ... \oplus \mathbf{s}_d,$$

   де d — порядок masking (рекомендується хоча б першого порядку);
3. **Shuffling:** Випадковізація порядку операцій [52];
4. **Подвійна перевірка:** Для захисту від fault injection виконувати re-encryption check двічі (!) незалежними одне від одного способами.
5. **Апаратні контрзаходи:** Використання HSM, secure enclaves.

Статус: Застосовна, вимагає ретельного захисту

Підсумовуючи вищезазначене, можна сказати що: side-channel атаки є найсерйознішою практичною загрозою для TiGER. Математична конструкція алгоритму стійка, але паршива реалізація може призводити до витоків секретів через фізичні канали і скомпрометувати загальну безпеку. Це вимагає впровадженню обов'язкових контрзаходів на рівні реалізації.

3.4 Порівняльні таблиці для TiGER

3.4.1 Порівняння атак на TiGER

Тут наведу підсумкові таблиці, думаю з них все і так зрозуміло :)

Тип атаки	Складність	Рівень загрози	Захист
BKZ	2^{144} (квант.)	Безпечно	Достатнє значення β
Brutforce	2^{393} (Grover)	Безпечно	Large key space
RLWR specific	?	Безпечно	Редукція до RLWE
Timing	Варіюється	Помірний	Const-time implementation
Power analysis	Варіюється	Помірний	Mask using
Fault injection	Варіюється	Безпечно	Implicit rejection

Таблиця 3.3: Слабкі місця TiGER

Тип атаки	Застосовність	Захист
Решіткові (BKZ)	Частково	Консервативні параметри забезпечують складність
Meet-LWE	Частково	Нереалістичний об'єм квантової пам'яті
Failure boosting	Ні	DFP $< 2^{-120}$ і implicit rejection
CCA	Ні	FO перетворення з re-encryption check
Side-channel	Так	Constant-time реалізація, masking, апаратні контрзаходи
Атаки на XEf	Теоретично	Подвійна корекція + implicit rejection + FO

Таблиця 3.4: Застосовність відомих атак до TiGER

Загальний висновок: Як бачимо, TiGER не має критичних математичних вразливостей. Основні його ризики пов'язані з side-channel атаками, які потребують обережної реалізації. Його "Cryptographic Core Architecture" є стійкою до всіх відомих теоретичних атак при заданих оптимальних параметрах.

3.4.2 Порівняння з іншими PQC алгоритмами

Алгоритм	**Решіткові**	**Meet-LWE**	**DFP захист**	**Side-channel***
Kyber	Стійкий	Частково	Низька DFP	Вразливий
Saber	Стійкий	Частково	Низька DFP	Вразливий
TiGER	Стійкий	Частково	IR + низька DFP	Вразливий
SMAUG-T⋆	Стійкий	Частково	IR + низька DFP	Вразливий

Таблиця 3.5: Порівняння стійкості TiGER з іншими постквантовими KEM

⋆ – криптопримітив, що оснований на TiGER;

* – Усі алгоритми вимагають constant-time реалізації, masking та апаратних контрзаходів

Підсумочок:

TiGER є **криптографічно стійким** постквантовим KEM алгоритмом, який ефективно протидіє відомим математичним атакам. Його математична конструкція базується на перевірених припущеннях складності (RLWE/RLWR) і дотримується балансу компактності та безпеки, а додаткові механізми (FO перетворення, implicit rejection, подвійна корекція помилок) забезпечують багаторівневий захист.

Основні ризики зосереджені більше у площині реалізації – side-channel атаки потребують обов'язкових контрзаходів. За умови дотримання рекомендацій щодо безпечної реалізації, TiGER може забезпечити надійний захист від квантових загроз для широкого спектру загроз на довгі роки. Злиття TiGER з SMAUG у алгоритм SMAUG-T та його визнання фіналістом KpqC підтверджує перспективність підходу та високу оцінку експертним криптографічним співтовариством.

P.S. Більш детально я розглядав цей алгоритм в своїй іншій лабораторній роботі.

Список використаних джерел

[1] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. 2nd ed. Boca Raton, FL: CRC Press, 2020. ISBN: 978-0815354369.

[2] Oded Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge, UK: Cambridge University Press, 2001. ISBN: 978-0521035361.

[3] Mihir Bellare et al. “Relations Among Notions of Security for Public-Key Encryption Schemes”. In: *Advances in Cryptology – CRYPTO ’98*. Vol. 1462. Lecture Notes in Computer Science. Springer, 1999, pp. 26–45. DOI: 10.1007/BFb0055718.

[4] Shafi Goldwasser and Silvio Micali. “Probabilistic Encryption”. In: *Journal of Computer and System Sciences*. Vol. 28. 2. 1984, pp. 270–299. DOI: 10.1016/0022-0000(84)90070-9.

[5] Alfred J. Menezes, Paul C. van Oorschot, and Scott Vanstone. *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 1996. ISBN: 978-0849385230.

[6] Moni Naor and Moti Yung. “Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks”. In: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*. STOC ’90. ACM, 1990, pp. 427–437. DOI: 10.1145/100216.100273.

[7] Charles Rackoff and Daniel R. Simon. “Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack”. In: *Advances in Cryptology – CRYPTO ’91*. Vol. 576. Lecture Notes in Computer Science. Springer, 1991, pp. 433–444. DOI: 10.1007/3-540-46766-1_35.

[8] Victor Shoup. “A Proposal for an ISO Standard for Public Key Encryption”. In: *IACR Cryptology ePrint Archive*. 2001. URL: https://eprint.iacr.org/2001/112.

[9] Silvio Micali, Charles Rackoff, and Bob Sloan. “The Notion of Security for Probabilistic Cryptosystems”. In: *SIAM Journal on Computing* 17.2 (1988), pp. 412–426. DOI: 10.1137/0217025.

[10] Danny Dolev, Cynthia Dwork, and Moni Naor. “Non-Malleable Cryptography”. In: *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*. STOC ’91. ACM, 1991, pp. 542–552. DOI: 10.1145/103418.103474.

[11] Mihir Bellare and Amit Sahai. “Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization”. In: *Advances in Cryptology – CRYPTO ’99*. Vol. 1666. Lecture Notes in Computer Science. Springer, 1999, pp. 519–536. DOI: 10.1007/3-540-48405-1_33.

[12] Mihir Bellare and Phillip Rogaway. “Optimal Asymmetric Encryption”. In: *Advances in Cryptology – EUROCRYPT ’94*. Vol. 950. Lecture Notes in Computer Science. Springer, 1994, pp. 92–111. DOI: 10.1007/BFb0053428.

[13] Eiichiro Fujisaki et al. “RSA-OAEP Is Secure under the RSA Assumption”. In: *Journal of Cryptology* 17.2 (2004), pp. 81–104. DOI: 10.1007/s00145-002-0204-y.

[14] Ronald Cramer and Victor Shoup. “A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack”. In: *Advances in Cryptology — CRYPTO ’98*. Vol. 1462. Lecture Notes in Computer Science. Springer, 1998, pp. 13–25. DOI: 10.1007/BFb0055717.

[15] Ronald Cramer and Victor Shoup. “Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack”. In: *SIAM Journal on Computing* 33.1 (2003), pp. 167–226. DOI: 10.1137/S0097539702403773.

[16] National Institute of Standards and Technology. *Module-Lattice-Based Key-Encapsulation Mechanism Standard*. Federal Information Processing Standards Publication FIPS 203. NIST, 2024. DOI: 10.6028/NIST.FIPS.203.

[17] Roberto Avanzi et al. “CRYSTALS-Kyber: Algorithm Specifications and Supporting Documentation”. In: *NIST Post-Quantum Cryptography Standardization*. Round 3 Submission. 2021. URL: https://pq-crystals.org/kyber/.

[18] Dan Boneh. “Twenty Years of Attacks on the RSA Cryptosystem”. In: *Notices of the American Mathematical Society* 46.2 (1999), pp. 203–213.

[19] Daniel Bleichenbacher. “Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1”. In: *Advances in Cryptology — CRYPTO ’98*. Vol. 1462. Lecture Notes in Computer Science. Springer, 1998, pp. 1–12. DOI: 10.1007/BFb0055716.

[20] Taher ElGamal. “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”. In: *IEEE Transactions on Information Theory* 31.4 (1985), pp. 469–472. DOI: 10.1109/TIT.1985.1057074.

[21] Yiannis Tsiounis and Moti Yung. “On the Security of ElGamal Based Encryption”. In: *Public Key Cryptography — PKC ’98*. Vol. 1431. Lecture Notes in Computer Science. Springer, 1998, pp. 117–134. DOI: 10.1007/BFb0054019.

[22] Oded Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge, UK: Cambridge University Press, 2004. ISBN: 978-0521830843.

[23] Shafi Goldwasser, Silvio Micali, and Ronald L. Rivest. “A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks”. In: *SIAM Journal on Computing* 17.2 (1988), pp. 281–308. DOI: 10.1137/0217017.

[24] Jason Chia, Ji-Jian Chin, and Sook-Chin Yip. “Digital signature schemes with strong existential unforgeability”. In: *F1000Research* 10 (Sept. 2021). ISSN: 2046-1402. DOI: 10.12688/f1000research.72910.1.

[25] Michel Abdalla et al. “From Identification to Signatures via the Fiat-Shamir Transform: Minimizing Assumptions for Security and Forward-Security”. In: *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology*. EUROCRYPT ’02. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 418–433. ISBN: 3540435530. URL: https://dl.acm.org/doi/10.5555/647087.715838.

[26] Mihir Bellare and Phillip Rogaway. “The Exact Security of Digital Signatures – How to Sign with RSA and Rabin”. In: *Advances in Cryptology – EUROCRYPT ’96*. Vol. 1070. Lecture Notes in Computer Science. Springer, 1996, pp. 399–416. DOI: 10.1007/3-540-68339-9_34.

[27] Claus-Peter Schnorr. “Efficient Signature Generation by Smart Cards”. In: *Journal of Cryptology*. Vol. 4. 3. 1991, pp. 161–174. DOI: 10.1007/BF00196725.

[28] David Pointcheval and Jacques Stern. “Security Proofs for Signature Schemes”. In: *Advances in Cryptology – EUROCRYPT ’96*. Vol. 1070. Lecture Notes in Computer Science. Springer, 1996, pp. 387–398. DOI: 10.1007/3-540-68339-9_33.

[29] National Institute of Standards and Technology. *Digital Signature Standard (DSS)*. Federal Information Processing Standards Publication FIPS 186-4. NIST, 2013. DOI: 10.6028/NIST.FIPS.186-4.

[30] Daniel R. L. Brown. “Generic Groups, Collision Resistance, and ECDSA”. In: *Designs, Codes and Cryptography* 35.1 (Apr. 2005), pp. 119–152. DOI: 10.1007/s10623-003-6154-z.

[31] National Institute of Standards and Technology. *Module-Lattice-Based Digital Signature Standard*. Federal Information Processing Standards Publication FIPS 204. NIST, 2024. DOI: 10.6028/NIST.FIPS.204.

[32] Joon-Yeon Cho et al. *TiGER: Tiny-polynomial based Ring-LWE/LWR scheme*. Round 3 Submission. Korean Post-Quantum Cryptography Competition, 2023. URL: https://kpqc.or.kr/images/pdf/TiGER.pdf.

[33] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. “On Ideal Lattices and Learning with Errors Over Rings”. In: *Advances in Cryptology – EUROCRYPT 2010*. Vol. 6110. Lecture Notes in Computer Science. Springer, 2010, pp. 1–23. DOI: 10.1007/978-3-642-13190-5_1.

[34] Abhishek Banerjee, Chris Peikert, and Alon Rosen. “Pseudorandom Functions and Lattices”. In: *Advances in Cryptology – EUROCRYPT 2012*. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 719–737. DOI: 10.1007/978-3-642-29011-4_42.

[35] National Institute of Standards and Technology. *Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process*. Call for Proposals. NIST, 2016. URL: https://csrc.nist.gov/pqc-standardization.

[36] Eiichiro Fujisaki and Tatsuaki Okamoto. “Secure Integration of Asymmetric and Symmetric Encryption Schemes”. In: *Journal of Cryptology*. Vol. 26. 1. Springer, 2011, pp. 80–101. DOI: 10.1007/s00145-011-9114-1.

[37] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. “A Modular Analysis of the Fujisaki-Okamoto Transformation”. In: *Theory of Cryptography Conference – TCC 2017*. Vol. 10677. Lecture Notes in Computer Science. Springer, 2017, pp. 341–371. DOI: 10.1007/978-3-319-70500-2_12.

[38] Daniele Micciancio and Shafi Goldwasser. *Complexity of Lattice Problems: A Cryptographic Perspective*. Vol. 671. The Kluwer International Series in Engineering and Computer Science. Springer, 2009. URL: https://share.google/TCxur9BglkiBHxIGC.

[39] Yuanmi Chen and Phong Q. Nguyen. “BKZ 2.0: Better Lattice Security Estimates”. In: *Advances in Cryptology – ASIACRYPT 2011*. Vol. 7073. Lecture Notes in Computer Science. Springer, 2011, pp. 1–20. DOI: 10.1007/978-3-642-25385-0_1.

[40] Anja Becker et al. “New directions in nearest neighbor searching with applications to lattice sieving”. In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2016, pp. 10–24. DOI: 10.1137/1.9781611974331.ch2.

[41] Erdem Alkim et al. “Post-quantum key exchange – A new hope”. In: (2016), pp. 327–343. URL: https://cryptojedi.org/papers/newhope-20151101.pdf.

[42] Thijs Laarhoven, Michele Mosca, and Joop van de Pol. “Finding shortest lattice vectors faster using quantum search”. In: *Designs, Codes and Cryptography* 77.2-3 (Apr. 2015), pp. 375–400. DOI: 10.1007/s10623-015-0067-5.

[43] Alexander May. “How to Meet Ternary LWE Keys”. In: *Advances in Cryptology – CRYPTO 2021*. Vol. 12826. Lecture Notes in Computer Science. Springer, 2021, pp. 701–731. DOI: 10.1007/978-3-030-84245-1_24.

[44] Jan-Pieter D’Anvers et al. “Decryption Failure Attacks on IND-CCA Secure Lattice-Based Schemes”. In: *Public-Key Cryptography – PKC 2019*. Vol. 11443. Lecture Notes in Computer Science. Springer, 2019, pp. 565–598. DOI: 10.1007/978-3-030-17259-6_19.

[45] Jan-Pieter D’Anvers et al. “Timing Attacks on Error Correcting Codes in Post-Quantum Secure Schemes”. In: *Proceedings of ACM Workshop on Theory of Implementation Security*. ACM, 2019, pp. 2–9. DOI: 10.1145/3338467.3358948.

[46] Casper von Berg. «On TiGER: the post quantum key encapsulation mechanism». Bachelor’s Thesis. Eindhoven University of Technology, 2024. URL: https://research.tue.nl/en/studentTheses/on-tiger/.

[47] Qian Guo, Thomas Johansson, and Alexander Nilsson. “A Key-Recovery Timing Attack on Post-Quantum Primitives Using the Fujisaki-Okamoto Transformation and Its Application on FrodoKEM”. In: *Advances in Cryptology – CRYPTO 2020*. Springer International Publishing, 2020, pp. 359–386. DOI: 10.1007/978-3-030-56880-1_13.

[48] Paul C. Kocher. “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems”. In: *Advances in Cryptology – CRYPTO ’96*. Vol. 1109. Lecture Notes in Computer Science. Springer, 1996, pp. 104–113. DOI: 10.1007/3-540-68697-5_9.

[49] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. “Differential Power Analysis”. In: *Advances in Cryptology – CRYPTO ’99*. Vol. 1666. Lecture Notes in Computer Science. Springer, 1999, pp. 388–397. DOI: 10.1007/3-540-48405-1_25.

[50] Daniel J. Bernstein. “Cache-timing attacks on AES”. In: *Technical report* (2005). URL: https://cr.yp.to/antiforgery/cachetiming-20050414.pdf.

[51] Jean-Sébastien Coron et al. “Higher-Order Side Channel Security and Mask Refreshing”. In: *Fast Software Encryption – FSE 2013*. Vol. 8424. Lecture Notes in Computer Science. Springer, 2014, pp. 410–424. DOI: 10.1007/978-3-662-43933-3_21.

[52] Matthieu Rivain and Emmanuel Prouff. “Provably Secure Higher-Order Masking of AES”. In: *Cryptographic Hardware and Embedded Systems – CHES 2010*. Vol. 6225. Lecture Notes in Computer Science. Springer, 2010, pp. 413–427. DOI: 10.1007/978-3-642-15031-9_28.