УДК 004.056.55 **DOI:10.30837/rt.2023.3.214.01**

С.О. КАНДІЙ, І.Д. ГОРБЕНКО, д-р техн. наук

АНАЛІЗ ДСТУ 8961:2019 У МОДЕЛІ КВАНТОВОГО ВИПАДКОВОГО ОРАКУЛА

Вступ

Особливістю сучасної криптографії є доказова безпека. Для сучасних криптографічних перетворень (за певних модельних припущень) існують формальні докази відсутності ефективних атак за умови складності декількох теоретико-числових проблем [1]. Для схем асиметричного шифрування та механізмів інкапсуляції ключів прикладами таких моделей безпеки є моделі на основі нерозрізнювальності – IND-CPA,IND-CCA1,IND-CCA2 [1].

Моделі на основі нерозрізнювальності є стандартним засобом для отримання формальних доказів безпеки, проте якщо в конструкції використовуються геш-функції, то часто отримати доказ існуючими засобами стає неможливо. Зазвичай, для подолання подібних труднощів використовується модель випадкового оракула [2], у межах якої геш-функції замінюються на ідеалізовані аналоги – випадкові оракули. Хоча така модель і не враховує специфічні атаки, що направлені на структуру геш-функцій, вона є стандартним засобом для оцінки безпеки в сучасній криптографії.

З розвитком квантових комп’ютерів з’явилися нові загрози, які модель випадкового оракула не враховує. Перші спроби побудувати модель квантового випадкового оракула [3] приносили лише обмежений успіх, оскільки звичні техніки доказу переставали працювати. В останні роки з’явилося багато нових технік [4], що дозволяють отримати докази, які раніше неможливо було отримати.

При формальному аналізі механізмів інкапсуляції ключів задача, зазвичай, розбивається на дві підзадачі: аналіз асиметричного перетворення, що лежить в основі, та аналіз перетворення, що робить з асиметричної схеми безпечний механізм інкапсуляції ключів [1]. В літературі було запропоновано багато таких перетворень. Зокрема, варто виділити перетворення Фуджісакі – Окамото [5], перетворення Дента [6], перетворення SXY [7]. Для них можливо знайти в літературі аналіз як у моделі випадкового оракула, так і у моделі квантового випадкового оракула.

Стандарт ДСТУ 8961:2019 [8] визначає асиметричне перетворення та механізм інкапсуляції ключів на основі NTRU [9]. Для отримання механізму інкапсуляції ключів використовується перетворення власної розробки, формального аналізу якого в літературі існує доволі мало. Структурно воно схоже на перетворення SXY, проте має деякі відмінності. У межах роботи будемо називати це перетворення як SkelyaTransform.

Мета роботи – аналіз перетворення SkelyaTransform у моделі квантового випадкового оракула. Наш аналіз ґрунтується на роботах [10], у яких проводився аналіз доволі схожого на SkelyaTransform перетворення.

1. Позначення

Для позначення предикатів використовується нотація $[[\cdot]]$. Якщо b є деяким твердженням, то предикат $[[b]]$ приймає значення 1, якщо b є істинним, та 0 інакше. Якщо змінна x приймає значення детермінованим чином, то використовується знак «=». Якщо змінна x приймає значення з деякого випадкового процесу, то використовується символ «$\leftarrow$». Для визначеної множини X позначення $x \leftarrow X$ означає, що змінна x приймає випадкове значення з рівномірного розподілу над X. Символом «==» позначатимемо перевірку на рівність аргументів. Ймовірність деякої події W надалі позначатимемо символом Pr[W], математичне очікування для деякого розподілу S надалі позначатимемо як E[S]. Для заданої множини X

вираз |X| означає потужність множини. Для числа x вираз |x| означає абсолютне значення. Для заданої схеми асиметричного шифрування MSpace є множиною допустимих повідомлень, RSpace є множиною випадкових значень, CSpace є множиною допустимих шифротекстів, KSpace є множиною ключів.

2. Модель безпеки

Схема асиметричного шифрування є трійкою алгоритмів (Gen,Enc,Dec), де: $Gen:1^{\lambda}\to(pk,sk)$ – поліноміальний ймовірнісний алгоритм генерації ключової пари. Приймає параметр безпеки 1^{λ} та повертає ключову пару (pk,sk), $Enc:(pk,m)\to C$ – поліноміальний ймовірнісний алгоритм шифрування. Приймає публічний ключ pk, повідомлення m та повертає шифротекст C, $Dec:(sk,C)\to\{m,\perp\}$ – детермінований поліноміальний алгоритм розшифрування. Приймає секретний ключ *sk*, шифротекст C та повертає повідомлення m у разі вдалої декапсуляції та символ помилки $\perp$ у разі виникнення помилок.

Алгоритм *Enc* є ймовірнісним, тобто він має деяку внутрішню випадковість *r*. У межах аналізу зручно виносити цю випадковість у аргумент функції і вважати *Enc* детермінованим алгоритмом, що має сигнатуру $Enc:(pk,m,r)\to C$. Цей прийом має назву дерандомізація і широко використовується у формальних доказах.

Схема асиметричного шифрування має властивість відновлення випадковості, якщо існує алгоритм RandomRecovery, що приймає у якості аргументів таємний ключ *sk*, повідомлення m, відповідний шифротекст $c=Enc(pk,m)$ та повертає значення *r*, що використовувалося під час шифрування.

Схема асиметричного шифрування має властивість відновлення повідомлення, якщо існує алгоритм MessageRecovery, що приймає у якості аргументів відкритий ключ *pk*, випадкове значення *r* та шифротекст *c*, що використовує *r* під час шифрування. Алгоритм MessageRecovery повертає повідомлення *m*, що зашифроване у шифротексті *c*, або символ помилки розшифрування, якщо шифротекст є некоректним.

Якщо схема асиметричного шифрування має властивість відновлення випадковості та властивість відновлення повідомлення, то надалі казатимемо, що схема асиметричного шифрування має властивість однозначного відновлення.

Згідно з визначенням [1] протокол інкапсуляції ключів є трійкою алгоритмів (Gen,Encaps,Decaps), де: $Gen:1^{\lambda}\to(pk,sk)$ – поліноміальний ймовірнісний алгоритм генерації ключової пари. Приймає параметр безпеки 1^{λ} та повертає ключову пару (*pk*,*sk*), $Encaps:pk\to(K,C)$ – поліноміальний ймовірнісний алгоритм інкапсуляції ключа. Приймає публічний ключ *pk* і повертає випадковий ключ *K* та його інкапсуляцію (шифротекст) *C*, $Decaps:(sk,C)\to\{K,\perp\}$ – детермінований поліноміальний алгоритм декапсуляції ключа. Приймає секретний ключ *sk* та інкапсуляцію ключа *C* і повертає ключ *K* у разі вдалої декапсуляції та символ помилки $\perp$ у разі виникнення помилок.

Стандартна модель безпеки схем асиметричного шифрування та механізмів інкапсуляції ключів ґрунтується на понятті обчислювальної нерозрізнювальності. Загальний принцип полягає у тому, що якщо для будь-якого ефективного алгоритму неможливо відрізнити шифротекст від випадкового значення при заданих модельних припущеннях, то схема шифрування вважається безпечною у межах цієї моделі.

Реалізується цей принцип через ігри (експерименти) між супротивником (що реалізований довільним набором алгоритмів) та іспитувачем. Іспитувач готує експеримент та викликає алгоритми супротивника, що мають визначений моделлю інтерфейс. Супротивник окрім переданих аргументів може робити запити до оракулів – алгоритмів, про реалізацію яких супротивник нічого не знає. Якщо супротивник A може робити запити до оракула O, то надалі позначатимемо це як A^{O}.

Від схем асиметричного шифрування при побудові механізмів інкапсуляції ключів вимагається безпека у моделі IND-CPA (Indistinguishability under Chosen-Plaintext Attacks) або у моделі OW-CPA (One-Wayness under Chosen-Plaintext Attacks). Відповідні експерименти зображені на рис. 1.

GAME OW-CPA:

1. $(pk, sk) \leftarrow KeyGen(1^{\lambda})$
2. $m^* \leftarrow MSpace$
3. $c^* \leftarrow Enc(pk, m^*)$
4. $m' \leftarrow A(pk, c^*)$
5. $return\ [[m' == m^*]]$

GAME IND-CPA:

1. $(pk, sk) \leftarrow KeyGen(1^{\lambda})$
2. $b \leftarrow \{0,1\}$
3. $(m_0^*, m_1^*) \leftarrow A_1(pk)$
4. $c^* \leftarrow Enc(pk, m_b^*)$
5. $b' \leftarrow A_2(pk, c^*)$
6. $return\ [[b' == b]]$

Рис. 1. Ігри OW-CPA та IND-CPA для схем асиметричного шифрування

Перевагу супротивника А у іграх IND-CPA та OW-CPA для схеми асиметричного шифрування PKE позначимо як $Adv_{PKE}^{OW-CPA}(A)$ та $Adv_{PKE}^{IND-CPA}(A)$ відповідно. Стандартним визначенням для переваги супротивника є:

$$\begin{aligned} Adv_{PKE}^{OW-CPA}(A) &= \Pr[OW-CPA(A) == 1] \\ \mathrm{Adv}_{PKE}^{IND-CPA}(A) &= \left|\Pr[IND-CPA(A) == 1] - \frac{1}{2}\right|. \end{aligned} \tag{1}$$

Від механізмів інкапсуляції ключів зазвичай вимагається безпека у моделі IND-CCA (Indistinguishability under Chosen-Ciphertext Attacks). На рис. 2 зображено гру для IND-CCA.

GAME IND-CCA:

1. $(pk, sk) \leftarrow KeyGen(1^{\lambda})$
2. $b \leftarrow \{0,1\}$
3. $(K_0^*, c^*) \leftarrow Encaps(pk)$
4. $K_1^* \leftarrow KSpace$
5. $b' \leftarrow A^{Decaps}(pk, c^*, K_b^*)$
6. $return\ [[b' == b]]$

$Decaps(c \neq c^*)$:

1. $K = KEM.Decaps(sk, c)$
2. $return\ K$

Рис. 2. Гра IND-CCA для механізмів інкапсуляції ключів

У грі IND-CCA супротивник може звертатися до оракула декапсуляції, який проводить декапсуляцію будь-якої інкапсуляції, окрім завдання, що було видано іспитувачем. Перевага супротивника А визначається наступним чином:

$$\mathrm{Adv}_{KEM}^{IND-CCA}(A) = \left|\Pr[IND-CCA(A) == 1] - \frac{1}{2}\right|. \tag{2}$$

Схема асиметричного шифрування у загальному випадку може мати помилки дешифрування, тобто для деяких правильно обчислених шифротекстів розшифрування може давати неправильний результат. Існують різні підходи до врахування помилок дешифрування. У межах роботи ми будемо слідувати [10]. Для оцінки ймовірності виникнення помилок дешифрування введемо величину

$$\delta_{wc} = \underset{(pk,sk)}{E}\left[\max_{m \in M} \Pr_{c}[Dec(sk, c) \neq m\right]. \tag{3}$$

Величина δ_{wc} характеризує ймовірність появи помилок дешифрування у найгіршому випадку.

Для того щоб схема асиметричного шифрування була безпечною, необхідно, щоб рівень помилок був незначним. Для оцінки складності отримання помилки дешифрування введемо гру COR-RO на рис. 3. У цій грі супротивник має доступ до деякого випадкового оракула G. Задача супротивника полягає у тому, щоб повернути список повідомлень. Якщо хоча б одне повідомлення викликатиме помилку дешифрування, то супротивник перемагає.

Game COR-RO:

1. $(pk, sk) \leftarrow PKE.KeyGen(1^{\lambda})$
2. $L_M \leftarrow A^G(sk, pk)$
3. $for\ m \in L_M$
4. $\quad c \leftarrow Enc(pk, m)$
5. $\quad$ if $Dec(sk, c) \neq m$
6. $\quad\quad$ return 1
7. return 0

Рис. 3. Гра COR-RO для схем асиметричного шифрування

Перевага супротивника A відповідно визначається як

$$Adv_{PKE}^{COR-RO}(A) = \Pr[COR - RO(A) = 1]. \quad (4)$$

У роботі [10] отриманий важливий результат щодо оцінки ймовірності появи помилок дешифрування.

Л е м а 1 . Якщо PKE є δ_{wc}-коректною схемою асиметричного шифрування, тоді для будь-якого супротивника A, що робить q_G квантових запитів до оракула G та повертає одне повідомлення, має місце нерівність

$$Adv_{PKE}^{COR-RO}(A) \leq 8 \cdot (q_G + 1)^2 \cdot \delta_{wc}. \quad (5)$$

Важливою лемою при доказі тверджень у моделях на основі нерозрізнювальності є так звана лема Union Bound.

Л е м а 2 (Union Bound, [6]). Нехай A,B та E – події у деякому просторі ймовірностей. Якщо $\Pr[A \mid \neg E] = \Pr[B \mid \neg E]$, то має місце нерівність $|\Pr[A] - \Pr[B]| \leq \Pr[E]$.

3. Перетворення SkelyaTransform

У межах роботи досліджується перетворення SkelyaTransform, яке визначено (у неявному вигляді) стандартом ДСТУ 8961. Наведемо формальний опис цього перетворення для довільної схеми асиметричного шифрування.

Нехай λ – параметр безпеки, PKE=(Gen,Enc,Dec) – деяка схема асиметричного шифрування, що використовує простір повідомлень MSpace, простір шифротекстів CSpace, простір випадковості RSpace і задано геш-функції:

$$\begin{aligned} &H : RSpace \to \{0,1\}^{\lambda} \\ &BPGM : MSpace \to RSpace. \\ &KDF : RSpace \to \{0,1\}^{\lambda} \end{aligned} \quad (6)$$

Перетворення SkelyaTransform задано наступним чином (рис. 4):

$SkelyaTransform.Gen(1^\lambda)$:	$SkelyaTransform.Encaps(pk)$:	$SkelyaTransform.Decaps(C=(C_1,C_2),sk)$):
1. Return $(pk,sk)=PKE.Gen(1^\lambda)$	1. $m \leftarrow_R M$	1. $m' = PKE.Dec(C_1, sk)$
	2. $r = BPGM(m)$	2. If $m' == \perp$ return $\perp$
	3. $C_1 = PKE.Enc(m, r, pk)$	3. $r' = BPGM(m)$
	4. $C_2 = H(r)$	4. $C_2' = H(r')$
	5. $K = KDF(r)$	5. $C_1' = PKE.Enc(m, r, pk)$
	6. $C = (C_1, C_2)$	6. If $C_1' == C_1 \&\& C_2' == C_2$
	7. return (C, K)	7. return $K = KDF(r)$
		8. return $\perp$

Рис. 4. Перетворення SkelyaTransform для довільної схеми асиметричного шифрування

4. Модель квантового випадкового оракула

Класичний випадковий оракул є функцією $H:\mathrm{X}\to Y$ (де $X=\{0,1\}^m$ та $Y=\{0,1\}^n$ для деяких m,n), яка обрана з рівномірного розподілу над множиною усіх можливих функцій Ω_H [4]. Квантовий випадковий оракул задається наступним оператором:

$$\mathrm{H}^{St}:|x,y\rangle\to|x,y\oplus H(x)\rangle. \tag{7}$$

Модель квантового випадкового оракула передбачає, що кожна геш-функція замінюється на квантовий випадковий оракул. Супротивник А може робити запити у суперпозиції до відповідних оракулів.

У класичній моделі випадкового оракула типовою стратегією доказу є показати, що супротивник А не може відрізнити значення випадкового оракула від випадкового, якщо А не робив раніше відповідного запиту до оракула. Проте, у квантовому випадку цю стратегію важко реалізувати, оскільки А може робити запити в суперпозиції і з деякою незначною ймовірністю отримати відповідне значення. Щоб оцінити ймовірність успіху, А необхідно оцінити наскільки важко витягти цю інформацію з запиту. Одним з перших рішень для цієї проблеми була OW2H Лема. Нижче наведений варіант цієї леми, який є зручним для доказу.

Л е м а 3 (OW2H Лема [10]). Нехай $O:\{0,1\}^n\to\{0,1\}^m$ – квантовий випадковий оракул та А – деякий квантовий алгоритм, що робить не більше q_O квантових запитів до О, який у свою чергу робить не більше $q_{O_1},q_{O_2},...,\mathrm{q}_{O_N}$ запитів до оракулів $O_1,O_2,...,O_N$. Нехай E^A є алгоритмом, що на запит x^* робить наступне: обирає випадковим чином число i з множини $\{1,...,q_O\}$, змінну y з $\{0,1\}^m$ та запускає $A^O(input)$, де $input$ є деякими даними, що отримані з (x^*,y) за допомогою довільного алгоритму $GenInput(x^*,y)$, допоки не відбудеться i-й запит. Після цього алгоритм E^A вимірює аргумент запиту у обчислювальному базисі та повертає результат виміру. Якщо А робить менше i запитів, то E^A повертає $\perp\notin\{0,1\}^n$. Тоді виконується нерівність

$$\begin{aligned}&|\Pr[\mathrm{OW2H(A)}\Rightarrow 1]-1/2]|\le q_O\cdot\sqrt{p_{FIND}}\Rightarrow\\&|\Pr[\mathrm{OW2H(A)}\Rightarrow 1\,|\,b=0]-\Pr[\mathrm{OW2H(A)}\Rightarrow 1\,|\,b=1]|\le 2q_O\cdot\sqrt{p_{FIND}},\end{aligned} \tag{8}$$

де гра OW2H визначена на рис. 5 та

$$p_{FIND}=\Pr[x'=x^*], \tag{9}$$

де ймовірність взята над усіма можливими значеннями $x^*\leftarrow\{0,1\}^n, x'\leftarrow E^A(x^*)$.

Game $OW2H$:

1. $x^* \leftarrow \{0,1\}^n$
2. $y_0^* = O(x), y_1^* \leftarrow \{0,1\}^m$
3. $b \leftarrow \{0,1\}$
4. $b' \leftarrow A^O(x^*, y_b^*)$
5. $return\ [[b' == b]]$

Рис. 5. Гра OW2H

5. Аналіз перетворення SkelyaTransform

Основний результат цієї роботи сформульовано в теоремі 1.

Т е о р е м а 1 . Нехай PKE є OW-CPA безпечною та δ_{wc}-коректною схемою асиметричного шифрування з властивістю однозначного відновлення, тоді SkelyaTransfom[PKE] є IND-CCA безпечним механізмом інкапсуляції ключів. Більш формально – для кожного квантового алгоритму A у грі IND-CCA проти KEM=SkelyaTransform[PKE], що робить $q_H, q_{BPGM}, q_{KDF}, q_D$ запитів до оракулів H,BPGM,KDF та оракула дешифрування, існує квантовий алгоритм B у грі OW-CPA проти схеми асиметричного шифрування PKE, для якого виконується нерівність

$$Adv_{KEM}^{IND-CCA}(A) \le (2 \cdot q_H + 2 \cdot q_D + q_{KDF}) \cdot \sqrt{Adv_{PKE}^{OW-CPA}(B)} + 8 \cdot (q_{BPGM} + q_D + 1)^2 \cdot \delta_{wc} \quad (10)$$

Д о к а з .

Перед тим, як перейти безпосередньо до доказу, розглянемо загальну структуру доказу. Для доказу використовується стандартна техніка “game hopping”. Для того щоб довести нерівність (10), розглядається серія ігор GAME0 – GAME6. Гра GAME0 відтворює гру IND-CCA. Кожна наступна гра спрощується у тому сенсі, що значення змінних замінюються на дійсно випадкові або змінна взагалі виводиться з використання. При цьому фіксується зміна переваги супротивника. Цей процес відбувається до тих пір, допоки не буде простого способу оцінити ймовірність перемоги супротивника у поточній грі.

Гра GAME0 зображена на рис. 6.

$GAME0$:

1. $(pk, sk) \leftarrow PKE.KeyGen(1^\lambda)$
2. $b \leftarrow \{0,1\}$
3. $m^* \leftarrow \{0,1\}^n$
4. $r^* = BPGM(m^*)$
5. $c_0^* \leftarrow PKE.Enc(pk, m^*, r^*)$
6. $K_0^* = KDF(r^*); K_1^* \leftarrow \{0,1\}^n$
7. $K^* = K_b^*$
8. $c_1^* = H(r^*)$
9. $b' \leftarrow A^{Decaps}(pk, c^* = (c_0^*, c_1^*), K^*)$
10. $return\ [[b' == b]]$

$Decaps((c_0, c_1) \neq (c_0^*, c_1^*))$:

1. $m' = PKE.Dec(c_0, sk)$
2. $if\ m' == \perp$
3. $\quad return\ \perp$
4. $r' = BPGM(m')$
5. $c_1' = H(r')$
6. $if\ C_0 == PKE.Enc(m', r', pk)\ \&\&\ c_1 == c_1'$
7. $\quad return\ K = KDF(r')$
8. $return\ \perp$

Рис. 6. Гра GAME0

Ця гра в точності повторює IND-CCA гру для SkelyaTransfom[PKE], тому має місце рівність

$$Adv_{KEM}^{IND-CCA}(A) = |\Pr[GAME0(A) = 1] - 1/2|.$$

У грі GAME1 замість використання оракула BPGM генеруватимемо випадкове значення r^*. В оракулі декапсуляції відповідно замість BPGM використовуватимемо функцію RandomRecovery:

$GAME1$:

1. $(pk, sk) \leftarrow PKE.KeyGen(1^{\lambda})$
2. $b \leftarrow \{0,1\}$
3. $m^* \leftarrow \{0,1\}^n$
4. $r^* \leftarrow \{0,1\}^n$
5. $c_0^* \leftarrow PKE.Enc(pk, m^*, r^*)$
6. $K_0^* = KDF(r^*); K_1^* \leftarrow \{0,1\}^n$
7. $K^* = K_b^*$
8. $c_1^* = H(r^*)$
9. $b' \leftarrow A^{Decaps}(pk, c^* = (c_0^*, c_1^*), K^*)$
10. $return\ [[b' == b]]$

$Decaps\big((c_0, c_1) \neq (c_0^*, c_1^*)\big)$:

1. $m' = PKE.Dec(c_0, sk)$
2. $if\ m' == \perp$
3. $\quad return\ \perp$
4. $r' = RandomRecovery(m', c_0)$
5. $c_1' = H(r')$
6. $if\ c_0 == PKE.Enc(m', r', pk)\ \&\&\ c_1 == c_1'$
7. $\quad return\ K = KDF(r')$
8. $return\ \perp$

Рис. 7. Гра GAME1

Розглянемо наскільки зміниться перевага супротивника при переході від гри GAME0 до GAME1. Позначимо як $DIFF$ подію, яка полягає у тому, що супротивник зможе відрізнити ігри GAME0 та GAME1. З Леми 2 маємо:

$$|\Pr[GAME0(A)=1]-\Pr[GAME1(A)=1]|\le \Pr[DIFF]$$
$$|\Pr[GAME0(A)=1]-1/2-\Pr[GAME1(A)=1]+1/2|\le \Pr[DIFF]$$
$$|\Pr[GAME0(A)=1]-1/2|-|\Pr[GAME1(A)=1]-1/2|\le \Pr[DIFF]$$
$$Adv_{KEM}^{IND-CCA}(A) \le Adv_{KEM}^{GAME1}(A)+\Pr[DIFF]$$

Різниця між іграми буде помітна якщо супротивник зможе знайти повідомлення, яке викликає помилку дешифрування. Тобто, якщо супротивник А сформує запит $(c_0 = PKE.\mathrm{Enc}(\mathrm{pk}, \mathrm{m}, \mathrm{BPGM}(\mathrm{m})), c_1)$, для якого $PKE.Dec(sk, c_0) \neq m$. Тоді можливо побудувати супротивника D у грі COR-RO, що ідеально симулює середовище для супротивника А. Супротивник D симулює гру IND-CCA та усі оракули для А, використовуючи алгоритм гри GAME1, та записує усі запити А до оракулів BPGM та оракула дешифрування. Для супротивника А симуляція буде ідеальною, поки не станеться подія $DIFF$. Застосовуючи Лему 1 та Лему 2, отримуємо, що ймовірність події обмежена $8\cdot(q_{BPGM}+q_D+1)^2\cdot\delta_{wc}$. Отже, маємо:

$$\mathrm{Adv}_{KEM}^{IND-CCA}(A) \le Adv_{KEM}^{GAME1}(A)+8\cdot(q_{BPGM}+q_D+1)^2\cdot\delta_{wc}.$$

У грі GAME2 замінимо K^* та c_1^* на дійсно випадкові значення:

$GAME2$:

1. $(pk, sk) \leftarrow PKE.KeyGen(1^{\lambda})$
2. $m^* \leftarrow \{0,1\}^n$
3. $r^* \leftarrow \{0,1\}^n$
4. $c_0^* \leftarrow PKE.Enc(pk, m^*, r^*)$
5. $K^* \leftarrow \{0,1\}^n$
6. $c_1^* \leftarrow \{0,1\}^n$
7. $b' \leftarrow A^{Decaps}(pk, c^* = (c_0^*, c_1^*), K^*)$
8. $return\ [[b' == b]]$

$Decaps\big((c_0, c_1) \neq (c_0^*, c_1^*)\big)$:

1. $m' = PKE.Dec(c_0, sk)$
2. $if\ m' == \perp$
3. $\quad return\ \perp$
4. $r' = RandomRecovery(m', c_0)$
5. $c_1' = H(r')$
6. $if\ c_0 == PKE.Enc(m', r', pk)\ \&\&\ c_1 == c_1'$
7. $\quad return\ K = KDF(r')$
8. $return\ \perp$

Рис. 7. Гра GAME2

Застосовуючи визначення переваги супротивника, отримуємо вираз

$$Adv_{KEM}^{IND-CCA}(A) \le \frac{1}{2}\cdot|\Pr[GAME1(A)=1|b=0]-$$
$$-\Pr[GAME1(A)=1|b=0]|+8\cdot(q_{BPGM}+q_D+1)^2\cdot\delta_{wc}$$
$$Adv_{KEM}^{IND-CCA}(A) \le 8\cdot(q_{BPGM}+q_D+1)^2\cdot\delta_{wc}+ \quad \cdot$$
$$+\frac{1}{2}\cdot|\Pr[GAME1(A)=1|b=0]-\Pr[\text{GAME 2(A)}=1]$$
$$+\Pr[\text{GAME 2(A)}=1]-\Pr[GAME1(A)=1|b=0]|$$

Звідки витікає

$$Adv_{KEM}^{IND-CCA}(A) \le 8\cdot(q_{BPGM}+q_D+1)^2\cdot\delta_{wc}+$$
$$\frac{1}{2}\cdot|\Pr[GAME1(A)=1|b=0]-\Pr[\text{GAME 2(A)}=1]|+\cdot$$
$$\frac{1}{2}\cdot|\Pr[\text{GAME 2(A)}=1]-\Pr[GAME1(A)=1|b=0]|$$

Для оцінки значень $|\Pr[GAME1(A)=1|b=0]-\Pr[\text{GAME 2(A)}=1]|$ та $|\Pr[\text{GAME 2(A)}=1]-\Pr[GAME1(A)=1|b=0]|$ можна застосувати лему OW2H. Якщо покласти $O(\cdot)=H(\cdot)$, то гра OW2H буде ідентичною до гри GAME1, за умови, що b=1 і до GAME2 якщо y є випадковим. Аналогічно, якщо покласти $O(\cdot)=H(\cdot)\times KDF(\cdot)$, то гра OW2H буде ідентичною до гри GAME1, за умови, що b=0 і до GAME2 якщо y є випадковим. З Леми 3 маємо нерівності:

$$|\Pr[GAME1(A)=1|b=0]-\Pr[GAME2(A)=1]|\le 2\cdot(q_{KDF}+q_H)\cdot\sqrt{\Pr[GAME3(A)=1]}$$
$$|\Pr[GAME1(A)=1|b=1]-\Pr[GAME2(A)=1]|\le 2\cdot(q_{KDF}+q_H+q_{KDF})\cdot\sqrt{\Pr[GAME4(A)=1]},$$

де GAME3, GAME4 зображені на рис. 9, де позначення E^A позначає запуск алгоритму A до тих пір, доки не буде обрано випадково чергу з запитів до відповідних геш-функцій, над якою потім робиться вимір для того, щоб отримати повідомлення m', відповідно до формулювання леми 3.

Для того щоб оцінити ймовірність успіху супротивника, у іграх GAME3, GAME4 змінимо оракул декапсуляції таким чином, щоб він не використовував секретний ключ, а відповідні ігри, що використовують змінений оракул декапсуляції, позначимо як GAME5,GAME6. При побудові нового оракула декапсуляції NewDecaps використаємо той факт, що квантовий випадковий оракул, до якого робиться q запитів, є невідрізнимим від випадкового полінома степені $2q$ над відповідним полем Галуа [10]. Відповідно, множина усіх значень r, для яких H(r)=d, може бути розглянута як множина коренів полінома H(X)-d. Новий оракул декапсуляції NewDecaps представлений на рис. 8. Замість таємного ключа для розшифрування повідомлення використовується множина значень r, що були вже запитані у оракула H.

Розглянемо як зміниться перевага супротивника від GAME3 до GAME5 та від GAME4 до GAME6. Нехай супротивник A робить запит до оракула декапсуляції з деяким шифротекстом $(c_0,c_1)\neq(c_0^*,c_1^*)$. Оракул декапсуляції Decaps для цього шифротексту може повернути ключ декапсуляції або символ помилки декапсуляції $\perp$.

Припустимо, що Decaps повертає $\perp$ для шифротексту (c_0,c_1), тоді, якщо NewDecaps не повертає $\perp$ у іграх GAME5-GAME6, то існує значення r для якого виконується $H(r)=c_1$.

Різниця між іграми буде, якщо для c_0, r існує повідомлення m, для якого $m = MessageRecovery(c_0, r) \neq \perp$. Проте, якщо таке m існує, то Decaps не буде повертати $\perp$, маємо протиріччя. Отже, таких m не існує і ігри в цьому випадку є такими, що не відрізняються.

$NewDecaps\big((c_0, c_1) \neq (c_0^*, c_1^*)\big)$:

1. if $\exists r \in Roots(H(X) - c_1) : PKE.Dec(sk, c_0) = m$
2. return $K = KDF(r)$
3. return $\perp$

Рис. 8. Оракул декапсуляції NewDecaps

GAME3:

1. $(pk, sk) \leftarrow KeyGen(1^\lambda)$
2. $m^* \leftarrow \{0,1\}^n$
3. $r^* \leftarrow \{0,1\}^n$
4. $K^* \leftarrow \{0,1\}^n$
5. $c_1^* \leftarrow \{0,1\}^n$
6. $c_0^* \leftarrow Enc(pk, m^*, r^*)$
7. $m' \leftarrow E^{A,H}(pk, (c_0^*, c_1^*), K^*)$
8. $return\ [[m' == m^*]]$

GAME4:

1. $(pk, sk) \leftarrow KeyGen(1^\lambda)$
2. $m^* \leftarrow \{0,1\}^n$
3. $r^* \leftarrow \{0,1\}^n$
4. $K^* \leftarrow \{0,1\}^n$
5. $c_1^* \leftarrow \{0,1\}^n$
6. $c_0^* \leftarrow Enc(pk, m^*, r^*)$
7. $m' \leftarrow E^{A,H,KDF}(pk, (c_0^*, c_1^*), K^*)$
9. $return\ [[m' == m^*]]$

$Decaps\big((c_0, c_1) \neq (c_0^*, c_1^*)\big)$:

1. $m' = PKE.Dec(c_0, sk)$
2. $if\ m' == \perp$
3. $\quad return\ \perp$
4. $r' = RandomRecovery(m', c_0)$
5. $c_1' = H(r')$
6. $if\ c_0 == PKE.Enc(m', r', pk)\ \&\&\ c_1 == c_1'$
7. $\quad return\ K = KDF(r')$
8. $return\ \perp$

Рис. 9. Ігри GAME3-GAME4

Припустимо, що Decaps не повертає $\perp$. Тоді існує деяке *r*, що є коренем H і NewDecaps повертає $K = KDF(r)$. Ігри і в цьому випадку є невідрізнимими і має місце рівність

$$\Pr[GAME3(A) = 1] = \Pr[GAME5(A) = 1]$$
$$\Pr[GAME4(A) = 1] = \Pr[GAME6(A) = 1].$$

Тож, задача звелася до оцінки складності ігор GAME5, GAME6. Для кожної з ігор можливо побудувати супротивників B_1, B_2 у грі OW-CPA проти PKE, які є обгорткою над A. Супротивник B_i симулює середовище для A наступним чином:

- Генерує випадкові значення K^* та c_1^*.
- Викликає $E^{A,Oracles}(pk, (c^*, c_1^*), K^*)$, де $Oracles = H$ для i=1 і Oracles=H,KDF для i=0.
- Повертає будь-що, що поверне $A^{E,Oracles}$.

Зрозуміло, що:

$$Adv_{PKE}^{OW-CPA}(B_1) = Adv_{KEM}^{GAME5}(A), Adv_{PKE}^{OW-CPA}(B_2) = Adv_{KEM}^{GAME6}(A)$$

Нехай супротивник B у грі OW-CPA проти PKE паралельно викликає B_1, B_2 для OW-CPA проти PKE. Зрозуміло, що $Adv_{PKE}^{OW-CPA}(B) = \min(Adv_{PKE}^{OW-CPA}(B_1), Adv_{PKE}^{OW-CPA}(B_2))$. Поєднуючи формули, маємо результат:

$$Adv_{KEM}^{IND-CCA}(A) \le (2 \cdot q_H + 2 \cdot q_D + q_{KDF}) \cdot \sqrt{Adv_{PKE}^{OW-CPA}(B)} + 8 \cdot (q_{BPGM} + q_D + 1)^2 \cdot \delta_{wc}$$

Що і треба було довести.

Висновки

У роботі отримано оцінки IND-CCA безпеки перетворення SkelyaTransform у моделі квантового випадкового оракула для довільних схем асиметричного шифрування з врахуванням помилок дешифрування, що мають властивість однозначного відновлення. На нашу думку, робота дозволить краще розуміти захищеність стандарту ДСТУ 8961:2019 від квантових атак. Головним недоліком нашого доказу є вимога властивості однозначного відновлення у схеми асиметричного шифрування. Така властивість виконується для переважної кількості реальних схем асиметричного шифрування, проте ця вимога є нестандартною. З іншої сторони, інші роботи, що присвячені аналізу асиметричних перетворень, також часто використовують схожу нотацію та додаткові нестандартні вимоги. Для складних перетворень, особливо у межах моделі квантового випадкового оракула, існує доволі мало результатів, що не використовують додаткових припущень щодо схем асиметричного шифрування. Тож, на нашу думку, отриманий результат є суттєвим для оцінки та розуміння безпеки діючого стандарту ДСТУ 8961:2019.

Список літератури

1. Горбенко І. Д., Горбенко Ю. І. Прикладна криптологія. Теорія. Практика. Застосування : монографія. Харків : Форт, 2012. 880 с.
2. Bellare S., Rogaway P. Random oracles are practical: a paradigm for designing eifficient protocols. ACM 1993
3. Boneh D., Dagdelen Ö., Fischlin M., Lehmann A., Schaffner C., Zhandry M. Random oracles in a quantum world // ASIACRYPT 2011. P. 41–69
4. Zhandry M. How to record quantum queries, and applications to quantum indifferentiability // CRYPTO 2019. P. 239–268.
5. Hofheinz D., Hovelmanns K., Kiltz E. A modular analysis of the fujisaki-okamoto transformation // Lecture Notes in Computer Science. 2017. Vol. 10677. P. 341–371.
6. Dent A. A Designer's Guide to KEMs. Cryptography and Coding. Cryptography and Coding, 2003. Vol 28. P. 29-56.
7. Saito T., Xagawa K., Yamakawa T. Tightly-Secure Key-Encapsulation Mechanism in the Quantum Random Oracle Model // EUROCRYPT 2018 . EUROCRYPT 2018. https://doi.org/10.1007/978-3-319-78372-7_17
8. ДСТУ 8961:2019. Інформаційні технології. Криптографічний захист інформації. Алгоритми асиметричного шифрування та інкапсуляції ключів. Чинний від 21.12.2019. Вид. офіц. Київ : УкрНДНЦ, 2019. 72 с.
9. Hoffstein J., Pipher J., Silverman H. NTRU: a ring based public key cryptosystem // Algorithmic Nuber Theory. Third International Symposium. 1998. P. 267–288.
10. Bindel N., Hamburg M., Hövelmanns K., Hülsing A., Persichetti E. Tighter proofs of CCA security in the quantum random oracle model // Dennis Hofheinz and Alon Rosen, editors. TCC 2019. P. 61–90.
11. Don J., Fehr S., Majenz C., Schaffner C. Online-Extractability in the Quantum Random-Oracle Model // EUROCRYPT 2022.

Надійшла до редколегії 08.08.2023

Відомості про авторів:

Кандій Сергій Олегович – Харківський національний університет імені В. Н. Каразіна, аспірант кафедри безпеки інформаційних систем і технологій, факультет комп'ютерних наук, АТ «Інститут Інформаційних технологій», молодший науковий співробітник, Україна; e-mail: sergeykandy@gmail.com; ORCID: https://orcid.org/0000-0003-0552-8341

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