УДК 004.056.5 **DOI:10.30837/rt.2025.2.221.07**

К.Є. ЛИСИЦЬКИЙ, PhD, І.В. ЛИСИЦЬКА, д-р техн. наук, І.М. ГАЛЬЦЕВА

ІДЕЯ ЗЛАМУ ГЕШ-ФУНКЦІЇ НА КВАНТОВІЙ ШВИДКОСТІ

Вступ

Квантові комп'ютери доволі швидко опановують світ. Так, є припущення експертів, що квантовий комп'ютер зможе впоратися зі 2048 бітовим шифруванням вже наприкінці 2030-х років.

Критично важливі елементи структури кібербезпеки державного управління, військового та промислового комплексу залишаються незмінними протягом десятиріч з перспективою їх подальшого використання. Стають зрозумілими спроби зі сторони зловмисників заволодіти великими об'ємами зашифрованих даних з надією їх накопичити та розшифрувати у майбутньому за допомогою квантових комп'ютерів – метод HNDL (Harvest Now, Decrypt Later – «Збери зараз, розшифруй пізніше»). Інформація, яку зашифрували з використанням криптографічних стандартів, що не є квантово-безпечними, може вважатися втраченою.

Вже існують так звані квантово-центричні суперкомп'ютери. Це досить складна обчислювальна архітектура, яка використовує при обчисленнях переваги паралельних обчислень з використанням одночасно квантових та класичних обчислень. Саме такі гібридні обчислювальні системи в найближчий час можуть бути основною загрозою для більшості класичних криптоалгоритмів.

До основних методів і задач криптоаналізу, що можуть бути вирішені за допомогою квантового комп'ютера та становлять загрозу для сучасних крипто перетворень, можна віднести наступні [1 – 3]:

1) алгоритм Гровера для пошуку в несортованій базі;

2) алгоритм факторизації Шора;

3) алгоритм Шора для розв'язку дискретного логарифму в скінченному полі;

4) алгоритм Шора для розв'язку дискретного логарифму в групі точок еліптичної кривої.

Саме через загрозу з боку квантових технологій у 2016 р. NIST розпочав конкурс щодо розробки квантово-безпечних стандартів з оприлюдненням вимог до майбутніх розробок [4 – 7]. Було проведено три раунди з відбору кандидатів на стандартизацію, розроблено проєкти стандартів та продовжено дослідження в четвертому раунді.

Постквантова криптографія (PQC) є головним пріоритетом національної безпеки розвинених держав, які готуються переходити до квантово-безпечних практик.

У травні 2021 р. Президент України підписав «Про Стратегію кібербезпеки України» [8].

У січні 2024 р. було опубліковано ключові ідеї квантової стратегії НАТО. Передбачено спрямовувати співпрацю НАТО з промисловістю для розвитку трансатлантичної екосистеми квантових технологій, одночасно готуючи НАТО до захисту від зловмисного використання квантових технологій.

Квантово-безпечна криптографія вже існує. Є декілька підходів до розробки алгоритмів, що не є вразливими до атак квантовими комп'ютерами. Це криптографія заснована на: алгебраїчних решітках, геш-функціях, математичних кодах, багатовимірній криптографії та ізогеніях суперсингулярних еліптичних кривих.

Постквантова криптографія на основі геш-функцій

Злам геш-функції означатиме наступне: виходячи з відповідного гешу знайти повідомлення, з якого цей геш було сформовано зі складністю менш, ніж O(n). Якщо це неможливо зробити, функція вважається криптографічно стійкою.

Вхідний набір даних найчастіше кодується у ASCII або двійковому форматі, а вихідний – у шістнадцятковому.

Квантовий комп'ютер за допомогою відомого алгоритму Гровера здатен знайти принаймні одне повідомлення, пов'язане з даним гешем зі складністю O(n/2), що значно краще, ніж O(n) з використанням класичного комп'ютера.

Багато прикладів використання алгоритму Гровера викликають розчарування: вони починають із кодування рішення в квантову схему Oracle, а потім запускають алгоритм Гровера, щоб знайти рішення, яке ми вже знаємо.

У [9] запропоновано цікаву ідею – закодувати саму геш-функцію в Oracle, а не рішення. У даному випадку алгоритм не знає рішення: він знає лише алгоритм гешування і геш-функцію, як і будь-який потенційний супротивник.

Для отримання справжнього криптографічного гешу, наприклад для алгоритму SHA2, знадобляться десятки тисяч квантових гейтів і реєстр із 128 кубітів (не рахуючи вражаючої кількості допоміжних кубітів) квантового комп'ютеру.

Для демонстрації ідеї запропонована вигадана геш-функція **(toy)**, яка працює від вхідного набору {0,1}**6 до цільового набору {0,1}**4, що використовує лише два квантові вентилі XOR і 6 кубітів. Ця функція не володіє добрими криптографічними властивостями і призначена лише показати цей підхід без втрати загальності [10].

```
def toy(message):
    message[0] = message[4] ^ message[5]
    message[2] = message[5] ^ message[0]
    return message
```

Кодування геш-функції в AWS Braket.

Згідно з ідеями [9, 10] виконуємо чотири кроки:

- обчислюємо геш (toy) з квантовими вентилями;
- позначаємо квантові стани, що відповідають заданому значенню геш-функції;
- uncompute toy () для скидання реєстру кубітів;
- посилюємо розмітку за допомогою оператора дифузії.

1. Комп'ютерна функція toy.

У AWS Braket можна представити функцію dream наступним чином:

```
@circuit.subroutine(register=True)
def quantum_hash():
    ocirc=Circuit()
    ocirc.ccnot(4,5,0).ccnot(5,0,2)
    return ocirc
```

@circuit.subroutine(register=True) – декоратор, який реєструє функцію quantum_hash як підпрограму у квантовому контексті;

ocirc = Circuit() – створюється новий квантовий цикл;

ocirc.ccnot(4, 5, 0) – елемент CCNOT (Toffoli-гейт), що застосовується до кубітів 4 і 5 (контрольні), впливає на кубіт 0 (цільовий);

ocirc.ccnot(5, 0, 2) – ще один Toffoli-гейт, де контрольні кубіти – 5 і 0, а цільовий – 2;

return ocirc – повертає побудовану квантову схему.

Для надання всього простору повідомлень завдовжки 6 бітів вводяться 6 кубітів, що охоплюють 2^6=64 можливих повідомлення. При цьому кубіти нумеруються від 0 до 5.

Перші 4 кубіти при квантовому обчисленні дають один з 2^4=16 можливих значень геш-функції. Останні 2 кубіти є частиною обчислення, що не рахуються у результаті, тому вони ігноруються при наступних кроках.

2. Позначаємо квантові стани, що відповідають заданому значенню геш-функції.

Значення геш-функції складається з чотирьох кубітів. Використовується вентиль CCCC-NOT для позначення відповідного квантового стану. Задіяно додатковий шостий кубіт, щоб звільнити результуючий вихід вентеля, який нам не потрібен.

3. uncompute dream() для скидання реєстру кубітів.

Зворотні обчислення виконуються просто обчисленням їх у зворотному порядку. Далі представлено uncompute квантового гешу:

```
@circuit.subroutine(register=True)
def rev_quantum_hash():
    ocirc=Circuit()
    ocirc.ccnot(5,0,2).ccnot(4,5,0)
    return ocirc
```

Необчислення, як відомо, – це техніка, яка використовується в оборотних схемах для очищення тимчасових впливів на допоміжні біти, щоб їх можна було використовувати повторно [11]. Необчислення є фундаментальним кроком в алгоритмах квантового обчислення.

Ці перші три етапи складають оракул квантового пошуку (рис. 1).

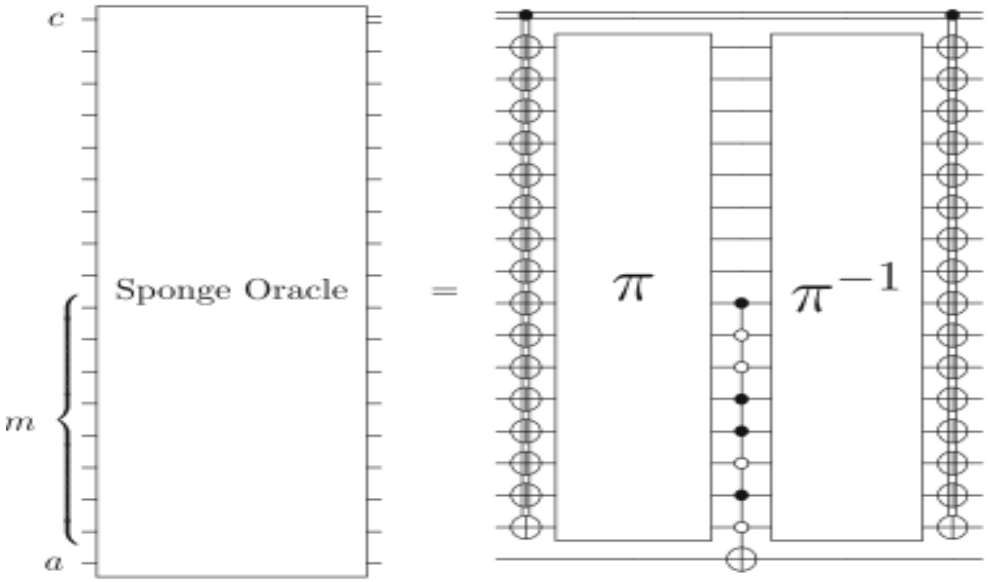

Рис. 1. Оракул квантового пошуку

У статті вони представлені наступним чином:

- "m" – можливі повідомлення в кубітах;
- "a" – це допоміжний кубіт, який використовується для вентилів CCCC-NOT;
- π – квантова геш-функція з першого кроку;
- шлюз CCCC-NOT, стиснутий між π та π^{-1}, налаштований на пошук гешу 10011010 (на рисунку у статті геш довжиною 8 біт, у нас в експерименті буде 4 біти) (крок 2);
- π^{-1} – uncompute квантова геш-функція (крок 3).

Далі регістр повертається до початкового стану і тоді сигнал готовий до посилення.

4. Посилення.

Використовується той же оператор дифузії, що запропоновано у статті (рис. 2).

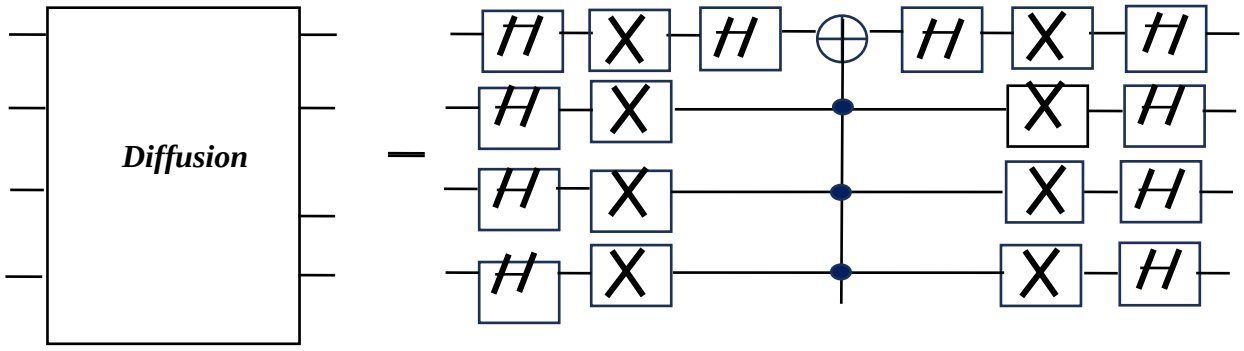

Рис. 2. Оператор дифузії

У нашому випадку потрібно лише 4 кубіти (геші мають 4 біти):

```
def diffuse(qbits=4):
    circ=Circuit()
    circ.h(np.arange(qbits))
    circ.x([0,1,2,3]).h([0]).cccnot(targets=[3,1,2,0]).h([0]).x([0,1,2,3])
    circ.h(np.arange(qbits))
    return circ
```

Це реалізація квантової дифузної операції (інверсія відносно середнього) – ключовий елемент алгоритму Гровера.

circ.h(np.arange(qbits)) застосовує Адамари до всіх кубітів (створює суперпозицію);

circ.x([...]) інвертує всі кубіти (підготовка до відображення);

circ.h([0]) → cccnot(targets=[3,1,2,0]) → circ.h([0])

реалізація мультикубітного контролюємого NOT (у даному випадку трьохконтролюємий Toffoli) з цільовим кубітом 0 – звичайно використовується як відображення відносно $|0\rangle$;

circ.x([...]) інвертування зворотно;

знову circ.h(np.arange(qbits));

завершення інверсії відносно середнього.

Ці чотири кроки ми можемо повторювати і більше одного разу. Але не занадто багато. Наша функція dream може обробляти 2^6=64 можливих повідомлення.

Якщо оцінити роботу класичного комп'ютера, то він може зламати геш-функцію після 64/2=32 спроб. Квантовий комп'ютер зробить це за $\sqrt{M}$ спроб, Тобто лише за 8 спроб, якщо M=1 і за 4 спроби, якщо M=4.

Нехай нам потрібно знайти повідомлення, що відповідає геш-функції 1111, що реалізовано як CCCC-NOT ([0,1,2,3,5]) в AWS Braket.

Коректність квантового пошуку у нашій спрощеній ситуації можна перевірити простим перерахуванням всіх повідомлень, які відповідають геш-функції 1111. Виявляється, чотири повідомлення дають геш-функцію 1111.

Це повідомлення 011110, 010110, 111110 і 110110.

Запускаємо програму AWS Braket двічі по 2 раунди кожного разу (2^2=4 спроби).

Ми повторюємо цю комбінацію 1111, щоб отримати середнє значення. Ось результати, найкращі збіги виділені жирним шрифтом:

{**'111110': 482**, **'110110': 416**, '110001': 4, '100000': 2, **'010110': 460**, '011111': 6, **'011110': 435**, '101111': 4, '111010': 7, '011101': 6, '111000': 4, '101101': 5, '101110': 4, '000000': 8, '110101': 3, '000100': 1, '100010': 5, '000110': 6, '010100': 4, '100111': 5, '010001': 3, '001100': 4, '111001': 5, '110110': 2, '101011': 3, '010000': 7, '001011': 2, '100011': 5, '011100': 2, '110000': 3, '010010': 1, '011011': 6, '111101': 5, '001110': 4, '101000': 4, '111111': 3, '011000': 3, '110011': 4, '011001': 4, '111011': 3, '110010': 7, '100110': 1, '111100': 4, '100100': 2, '110111': 3, '001000': 2, '001010': 4,

'100101': 5, '000101': 5, '100001': 3, '101100': 1, '011010': 3, '001111': 3, '000001': 2, '101001': 2, '000010': 4, '010101': 2, '010011': 3, '000111': 1, '101010': 1, '001101': 2}

Висновки за результатами експерименту.

Після чотирьох спроб ми отримуємо наступні результати:

- ймовірність знаходження квантовою схемою повідомлення 111110 становить 48,2 %;
- ймовірність знаходження квантовою схемою повідомлення 010110 становить 46 %;
- ймовірність знаходження квантовою схемою повідомлення 011110 становить 43,5 %;
- ймовірність знаходження квантовою схемою повідомлення 110110 становить 41,6 %.

Тобто, ймовірність знайти за допомогою квантового комп'ютера хоча б одне з повідомлень, якому відповідає геш-функція 1111, становить близько 90 %.

Щодо класичного комп'ютера, то ймовірність знайти одне повідомлення становить близько 5 %, знайти хоча б одне повідомлення – близько 23 %.

Тобто квантові комп'ютери значно прискорюють інверсію геш-функцій і є вагомі причини хвилюватися щодо криптографічної стійкості примітивів, що засновані на комбінаториці. А це не тільки геш-функції.

Висновки

Алгоритм Гровера теоретично знижує стійкість геш-функцій до атак на прообраз та колізій. Це означає, що для підтримки аналогічного рівня безпеки в квантову еру розмір виходу геш-функцій може знадобитися збільшити вдвічі. Існують теоретичні дослідження щодо побудови квантових схем для обчислення геш-функцій та їх обернення, але експериментальні реалізації цих схем на сьогоднішніх квантових комп'ютерах є дуже обмеженими і стосуються лише спрощених версій геш-функцій.

Розробка достатньо потужних і стабільних квантових комп'ютерів, здатних виконати такі складні обчислення, є серйозним науково-технічним викликом.

Проте, дослідження в галузі постквантової криптографії активно розвиваються, і вже існують перспективні криптографічні алгоритми, які, як вважається, будуть стійкими до атак як класичних, так і квантових комп'ютерів, включаючи геш-функції.

Список літератури:

1. Горбенко І. Д., Горбенко Ю. І. Прикладна криптологія : підручник. 2-ге вид. Харків : Форт, 2013. 878 с.

2. Горбенко Ю.І. Методи побудування та аналізу криптографічних систем : моногр. Харків : Форт, 2015. 959 с.

3. Gorbenko I.D. Methods of building general parameters and keys for NTRU Prime Ukraine of 5th–7th levels of stability. Product form / I.D. Gorbenko, O.G. Kachko, Yu.I. Gorbenko, I.V. Stelnik, S.O. Kandyi, M.V. Yesina // Telecommunications and Radio Engineering, 2019. Vol. 78. Is. 7. P. 579–594. Режим доступу: 10.1615/TelecomRadEng.v78.i7.30.

4. NIST IR 8413. Status Report on the Third Round of the NIST PostQuantum Cryptography Standardization Process. July 2022 (Updated 9/26/2022). [Електронний ресурс]. Режим доступу: 10.6028/NIST.IR.8413-upd1.

5. NIST IR 8105. Report on Post-Quantum Cryptography. April 2016. [Електронний ресурс]. Режим доступу: 10.6028/NIST.IR.8105.

6. NIST IR 8240. Status Report on the First Round of the NIST PostQuantum Cryptography Standardization Process. January 2019.[Електронний ресурс]. Режим доступу: 10.6028/NIST.IR.8240.

7. NIST IR 8309. Status Report on the Second Round of the NIST PostQuantum Cryptography Standardization Process. July 2020. [Електронний ресурс]. Режим доступу: 10.6028/NIST.IR.8309.

8. Про рішення Ради національної безпеки і оборони України від 14 травня 2021 року «Про Стратегію кібербезпеки України»: Указ Президента України; Стратегія від 26.08.2021 № 447/2021. Режим доступу: https://zakon.rada.gov.ua/laws/show/447/2021#n12.

9. Quantum Search for Scaled Hash Function Preimages Sergi Ramos-Calderer12, Emanuele Bellini1, José I. Latorre123, Marc Manzano1 and Victor Mateu1 arXiv:2009.00621v1 [quant-ph] 1 Sep 2020 1 Technology Innovation Institute, United Arab Emirates.

10. Bertoni Guido, Daemen Joan, P Michaël, and VA Gilles. Cryptographic sponge functions, 2011.

11. Aaronson Scott, Grier Daniel, Schaeffer Luke (2015). The Classification of Reversible Bit Operations. arXiv:1504.05155 [quant-ph].

Надійшла до редколегії 19.02.2025

Відомості про авторів:

Лисицький Костянтин Євгенійович – PhD, Харківський національний університет імені В. Н. Каразіна, доцент кафедри математичного моделювання і аналізу даних, навчально-науковий інститут комп'ютерних наук та штучного інтелекту; Україна; e-mail:constantin.lisickiy@gmail.com; ORCID: https://orcid.org/0000-0002-7772-3376

Лисицька Ірина Вікторівна – д-р техн. наук, професор, Харківський національний університет імені В.Н. Каразіна; професор кафедри кібербезпеки інформаційних систем, мереж і технологій, навчально-науковий інститут комп'ютерних наук та штучного інтелекту, Харківський національний університет радіоелектроніки, професор кафедри безпеки інформаційних технологій; Україна; e-mail: ivlisitska@karazin.ua; ORCID**:** https://orcid.org/0000-0001-6758-9516

Гальцева Ірина Михайлівна – канд. техн. наук, доцент, Харківський національний університет імені В.Н. Каразіна, старший викладач кафедри кібербезпеки інформаційних систем, мереж і технологій, навчально-науковий інститут комп'ютерних наук та штучного інтелекту; Україна; e-mail:irina.galceva@karazin.ua