

ЗАСТОСУВАННЯ КУБІЧНИХ АТАК ДО БЛОКОВИХ ТА ПОТОКОВИХ ШИФРІВ

І. В. Свічкарьов^{1, а}

¹ Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»,
Фізико-технічний інститут

Анотація

У даній роботі розглядається застосування та покращення кубічних атак до блокових та поточкових шифрів на основі атак з використанням відкритого тексту.

Ключові слова: кубічні атаки, атаки за побічними каналами, вага Хеммінга, тести лінійності

Вступ

Більшість криптографічних схем можуть бути описані за допомогою полінома над полем $GF(2)$, який містить деякі секретні змінні (наприклад, секретний ключ) і публічні змінні (наприклад, відкритого тексту, зашифрованого тексту, IV тощо). Кубічні атаки є свого роду алгебраїчними атаками, запропоновані Дінуром та Шаміром на EUROCRYPT 2009. [1] Ідея таких атак – це генерування достатньої кількості тісно пов'язаних рівнянь низького степеню, маніпулюючи публічними змінними. Супротивник може потім вирішити систему згенерованих рівнянь для відновлення секрету криптосистеми.

1. Короткий опис кубічних атак

Для кубічної атаки передбачається криптосистема, яка приймає на вхід відкриті та секретні параметри. Передбачається, що зломисник не знає структури роботи криптосистеми, але отримує доступ до двох речей:

- 1) Симулятор (шифратор), який приймає пари секретних і відкритих входів і виробляє біти виходу.
- 2) Оракул, який містить приховану таємницю. І коли подаються відкриті параметри, виробляється вихідне значення (біт шифртексту) на виході.

Кубічна атака дозволяє атакуючому вивести таємницю оракула на основі атаки з використанням відкритого тексту. Загалом, кубічна атака виконується в два етапи: попередня обробка та онлайн-фаза.

Представимо многочлен у вигляді суми доданків:

$$p(x_1, \dots, x_n) \equiv t_I \cdot p_{S(I)} + q(x_1, \dots, x_n),$$

де $p_{S(I)}$ – називають суперполіномом (superpoly) множини I , який не містить спільних змінних із t_I і кожен член в $q(x_1, \dots, x_n)$ не містить принаймні одну змінну із I , при чому ніякий із членів в $q(\bar{x})$ не ділиться на t .

Множина I визначає k -розмірний булевий куб C_I . Наступне визначення знадобиться для подальшого розгляду:

Максимальним членом (maxterm) полінома p – називається такий член $t_I : \deg(p_{S(I)}) \equiv 1$, тобто лінійний і не константа.

На етапі попередньої обробки, мета полягає в тому, щоб знайти максимальні члени, які відповідають суперполіномам і отримати алгебраїчну структуру цих суперполіномів. Після того, як достатня кількість лінійно незалежних суперполіномів було виявлено, вони можуть зберігатися і використовуватися у всіх наступних онлайн-фазах атаки на цю криптосистему.

Більш детальну інформацію про знаходження суперполіномів та структуру кубічних атак можна знайти у [1],[2],[3],[4]

2. Теоретична складність атаки

Розглянемо поліном степені d із n секретними і m відкритими входними бітами. Складність обчислення значень n суперполіномів оцінюється щонайбільше $2^{d-1}n$ запитами до оракула. Складність рішення системи лінійних рівнянь $\mathcal{O}(n^2)^2$.

Таким чином, загальна складність онлайн-атаки $\mathcal{O}(2^{d-1}n) + \mathcal{O}(n^2)$.

Що стосується етапу попередньої обробки, він складається з оцінки кубів (maxterm's cube), кожна з яких оцінюється $\mathcal{O}(2^{d-1})$.

Загальна складність оцінки нелінійних рівнянь степені D використовуючи алгоритми, які описані у [3], наступна:

$$\kappa n 2^{d-D-1} + \sum_{i=0}^D 2^{d-D-i} C_i^N,$$

де N – кількість секретних змінних яка оцінюється за алгоритмом, який наведений у [3], а κ – кількість

²За умови, що матриця (matrix of superpolys) лінійних рівнянь не вироджена, яка може бути доповнена ще кількома суперполіномами, якщо це необхідно.

тестів для знаходження секретних змінних. Для більшої впевненості у правильності роботи алгоритму $\kappa = 300$.

Слід зазначити, що значення d , як правило, не відоме до виконання атаки, так що верхня межа існує, але злоумисник не обов'язково знає яка вона.

3. Практична складність атаки

Тут вже на відміну від теоретичних засад виникають такі проблеми, як:

- 1) пошук кубів;
- 2) недостатня кількість знайдених поліномів;
- 3) гарний\поганий оракул;
- 4) пошук моделі витоку.

Перша проблема: Якщо казати про блокові шифри, де використовують достатню кількість раундів, щоб степінь поліному зростала експоненційно, то пошук кубів у такому випадку стає дуже складним і немає ефективного їх пошуку. Тому для цього використовують не повний перебір, а рандомний вибір (random walk), який завдяки зміні розміру кубу дає змогу, на відміну від повного перебору, знайти більшу кількість кубів за менший час.

Також при пошуку можливі варіанти *розширеної атаки*, де вже можливе доповнення кубів однією додатковою змінною або об'єднання двох кубів і подальша їх перевірка на коректність, що і було використано при реалізації атаки на шифр.

Друга проблема:

Коли нам не вистачає лінійних поліномів для рішення системи, то можливе її доповнення нелінійними, що буде ускладнювати роботу. Так як рішення великих систем поліноміальних рівнянь з багатьма змінними, навіть якщо вони є тільки квадратичними, є так званою *NP*-повною задачею. На цьому заснована стійкість багатьох криптографічних алгоритмів. Використання алгоритмів на основі базисів Грьобнера по відношенню до поліномів з числом невідомих більше 100 вимагає швидко зростаючих витрат пам'яті і ймовірно не може дати практичних результатів у криптоаналізі.

Спроби лінеаризації і побудови системи лінійних рівнянь за методом виключень Гаусса також не дають результату. Так для системи із 128 рівняннями у полі $GF(2)$ на поліномах при $d = 16$ і $n = 128$, кількість членів розв'язуваних рівнянь зростає до $n^d = 2^{128}$, що робить дану атаку непрактичною в порівнянні з простим перебором. Покращення даної техніки в вигляді *XL* і *XSL* алгоритмів стали вагомими досягненнями у криптоаналізі, але все ще залишалися непрактичними в більшості випадків.

Тому у випадках часто було легше довіритись власним навичкам і виражати одні змінні через інші, і підставляти в рівняння для пошуку бітів ключа, якщо це можливо.

Третя проблема:

Оракул може довго працювати, що дуже ускладнить нам роботу. Взагалі кажучи коли ми шукаємо куби, нам потрібно перевіряти суперполіном на лінійність, що потребує зайвих затрат. Використовувався

ймовірнісний *BLR* тест [5],[6], який вимагав 100 запитів для зменшення похибки. Тут також важливу роль грав генератор випадкових чисел, якщо він був не зовсім випадковим, то тест помилявся с високою ймовірністю.

Тому треба мати якийсь гарний алгоритм (оракул) на перевірку степеня полінома. Так як більша їх частина при перевірці кубів буде відкинута як константні, то гарно було б мати як в квантовій моделі алгоритм Дойча-Йожи, який працює набагато швидше за класичну модель.

Четверта проблема: Як вже було зазначено, із зростанням кількості раундів шифрування, степінь поліному зростає експоненційно. Тому важливим фактором для успішної атаки є модель витоку, такі як: окремий біт ваги Хеммінга внутрішнього стану після r -ого раунду або біт ваги від окремого байту внутрішнього стану. Тобто одним із варіантом є отримання парності/непарності від ваги.

Дуже важливим моментом є вибір раунду. Якщо номер раунду r досить малий, то кількість відновлених бітів ключа буде малою і складність відновлення ключа буде досить високою. Якщо r досить великий, то максимальні члени будуть відповідати поліномам з більшою кількістю ключових бітів, а степінь поліномів і їх кількість будуть рости експоненційно.

4. Пропонована атака на шифр SPECK

4.1. Короткий опис SPECK

SPECK – блоковий шифр який оптимізовано для роботи у програмній реалізації та належить до сімейства ARX (add-rotate-xor) систем.

SPECK має декілька варіантів реалізації з використанням різних розмірів блоків та ключів. Блок завжди складається із двох слів, але слова можуть бути розмірів: 16, 24, 32, 48 або 64 біти. Відповідний ключ 2, 3 або 4 слова. Кількість раундів залежить від розмірів блоку та ключа.

Більш детальну інформацію можна знайти у [7].

4.2. Пропонована модель атаки

Було застосовано кубічну атаку із використанням моделі витоку на шифр SPECK із 64-бітним ключем та 32-бітним блоком. В результаті, якщо знімати перший біт ваги Хеммінга від стану після 4-ого раунда, то можна отримати систему лінійних рівнянь, за допомогою якої можна відновити біти ключа.

4.3. Результати

У результаті атаки було знайдено 6 кубів розміру 1, 15 – розміру 2, 20 – розміру 3, 16 – розміру 4, 6 – розміру 5 та 1 куб розміру 6. Для знаходження правої частини усіх 64-ох рівнянь знадобилося $6 \cdot 2 + 15 \cdot 2^2 + 20 \cdot 2^3 + 16 \cdot 2^4 + 6 \cdot 2^5 + 2^6 \approx 2^{9.54}$ вибраних відкритих текстів.

У таблиці 1 знаходяться поліноми та відповідні їм куби. Так як поліноми вийшли великими, то їх подання буде подаватися у такому вигляді: поліном

Табл. 1. Приклади суперполіномів

№	Superpoly equations	Cube indexes
1	0000100100100111000010001000010001111111111010111000001100	31
2	1101011101010100100011001110010010000000011101011101110000011000	30
3	1010100011101010010000000101011100011110000001001011100000010000	28,29,30,31
4	1010100011101010010000000101011100011110000001001011100000010000	29
⋮	⋮	⋮
38	1110101010101110100000011011110010000101011001101000000000000000	26,29,31
39	1001110001111010110010011001011100111011011111001010000000000000	26,29,30
40	1111000110001011010010111100100111011101001100001001000000000000	26,29,30,31
41	0110110111011100010000101111011111111110111100010000000000000000	26,28
42	0001101101011000111011111000010110011000111101111000000000000000	26,28,31
43	0001101000010110010111110101110100100000010010100000000000000000	26,28,30
44	1011100010110111010010000010110100111000100000111000000000000000	27,28,29,30,31
⋮	⋮	⋮
61	0011000101001001010010010100000110010110110110110100000110000101	26
62	1001110001111010110010011001011100111011011111001010000000000000	26,29,30
63	1111000110001011010010111100100111011101001100001001000000000000	26,29,30,31
64	0111011001000000001001101100111000110010111101010000000000000000	26,27,28,29,30,31

$1 + x_1 + x_2 + x_6$ буде представлений у вигляді вектору 1110001.

Висновки

Таким чином було проведено аналіз блокових та поточкових шифрів на їх уразливість до кубічних атак. Взагалі, якщо не мати ніяких побічних каналів, то чисті кубічні атаки не несуть ніякої небезпеки для сучасних криптосистем. Звісно ж таких ідеальних умов не існує. Тому такий тип атак гарно буде доповнювати інші методи криптоаналізу.

Оскільки дана атака впершу чергу зачіпає полегшені варіанти шифрів, то добре спроектовані шифри, стійкі до широкого класу раніше відомих атак і володіють запасом міцності, поки залишилися стійкими і до цієї атаки.

Злам сучасних блокових шифрів важкий, так як вони мають високу степінь алгебраїчної складності, яка росте експоненційно із ростом числа раундів (з досягненням певного максимального значення $(n + m)$). Однак будь-який блоковий шифр, степінь складності якого менше ніж розмір ключа і відкритого тексту може бути гарантовано атакований.

У разі поточкових шифрів атака зачіпає впершу чергу теж тільки полегшені варіанти. Наприклад відомо, що регістр зсуву зі зворотним зв'язком – нестійка лінійна функція. Якщо використати нелінійний фільтр на S -блоках, з нього можна отримати простий і швидкий поточковий шифр. Тепер відомо, що він буде нестійкий до кубічної атаки.

Перелік використаних джерел

1. Dinur, I. and Shamir, A. Cube Attacks on Tweakable Black Box Polynomials. In A. Joux (Ed.), Advances

in Cryptology - EUROCRYPT 2009, Vol. 5479, pp. 278-299, Springer Berlin Heidelberg, 2009. — Access mode: <http://eprint.iacr.org/2008/385.pdf>.

2. Salam, Md Iftekhar, Bartlett, Harry, Dawson, Ed, Pieprzyk, Josef, Simpson, Leonie, Wong, Kenneth Koon-Ho. Investigating Cube Attacks on the Authenticated Encryption Stream Cipher ACORN, 2016. — Access mode: <http://eprint.iacr.org/2016/743.pdf>.
3. Shekh Faisal Abdul-Latip, Mohammad Reza Reyhanitabar, Willy Susilo. Extended Cubes: Enhancing the Cube Attack by Extracting Low-Degree Non-Linear Equations.
4. Shekh Faisal Abdul-Latip— 2012. — Access mode: <https://goo.gl/jaC6vA>.
5. M. Blum, M. Luby, and R. Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. Journal of Computer and System Sciences, vol 47(1993), pp. 549-595.
6. P. Mroczkowski, J. Szmids. Cube Attack on Stream Cipher Trivium and Quadraticity Tests. — 2010. — Access mode: <https://eprint.iacr.org/2010/580.pdf>.
7. Beaulieu, Ray; Shors, Douglas; Smith, Jason; Treatman-Clark, Stefan; Weeks, Bryan; Wingers, Louis «The SIMON and SPECK Families of Lightweight Block Ciphers». — Access mode: <http://eprint.iacr.org/2013/404.pdf>.