УДК 004.056.55

ШЛЯХИ ЗАСТОСУВАННЯ АТАКИ ЗА МАКСИМАЛЬНИМ СТЕПЕНЕМ ОДНОЧЛЕНА ДО ІНІЦІАЛІЗАЦІЇ КЛЮЧА ПОТОКОВОГО ШИФРУ «СТРУМОК»

О. Ю. Ковалевський[1]

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

Анотація

В роботі розглянуто атаку за максимальним степенем одночлена до ініціалізації ключів потокових шифрів та покращений жадібний алгоритм вибору бітів для цієї атаки. Запропоновано алгоритм побудови даної атаки на шифр «Струмок».

Ключові слова: потоковий шифр «Струмок», атаки ініціалізації, MDM-атака

Вступ

Потоковий шифр «Струмок» описано у діючому стандарті симетричного потокового шифрування в Україні. Актуальність роботи обумовлена тим, що шифр «Струмок» є мало дослідженим, зокрема не досліджена стійкість шифру до атаки за максимальним степенем одночлена.

1. Атака за максимальним степенем одночлена

Атака за максимальним степенем одночлена, яку також називають *MDM-атакою* (від англ. maximum degree monomial – моном максимального степеня), є статистичним тестом, який дозволяє дослідити здатність генератора ключового потоку виробляти моном максимального степеня.

Сучасні потокові шифри побудовані таким чином, що вони не можуть одразу після задання секретного ключа K і вектора ініціалізації IV створювати вихідну гаму, бо в такому разі функціональні зв'язки бітів вихідної гами із бітами секретного ключа будуть більш простими. Тому сучасні потокові шифри мають процедуру ініціалізації, яка перемішує певним чином біти вектора ініціалізації. Така процедура виконується якусь певну кількість ітерацій. MDM-тест було спеціально розроблено та запропоновано у роботі [3], щоб досліджувати наявність моному максимального степеня у алгебраїчній нормальній формі булевої функції генерації першого біту [3, 5] ключового потоку після ініціалізації шифру. За перетворенням Ріда-Мюллера коефіцієнт монома максимального степеня може бути знайдений за допомогою операції XOR для усіх можливих записів у таблиці істиності булевої функції першого біта гами:

$$\bigoplus_{x\in\{0,1\}^b} z_0(x), \text{ де } b \text{ – бітова довжина } x.$$

Проблема полягає у тому, що бітова довжина IV у більшості сучасних потокових шифрів перевищує 256 бітів, іноді – у кілька разів, тобто перебрати усі ці значення неможливо на сучасних обчислювальних машинах. Тому у роботі [3] було запропоновано використовувати не увесь простір $\{0,1\}^b$, а якусь його підмножину S.

Алгоритм MDM-тесту приймає на вхід алгоритм шифрування та числа k і P, де k – це кількість бітів у множині $|S|$, а P – це кількість многочленів, які буде згенеровано. На виході алгоритм каже чи можна відрізнити шифр від випадкового. Алгоритм MDM-тесту можна описати такими логічними кроками.

1) Обрати множину S, $|S| = k$, S – це множина бітів, які будуть змінюватися. Біти, які не увійшли до цієї множини, покладемо рівними константі, визначимо їх як фіксований набір.
2) Для усіх можливих значень вектора ініціалізації IV, які можна отримати із бітів множини S, ініціалізувати шифр. Отримати перший біт гами. За перетворенням Ріда-Мюллера обчислити коефіцієнт перед мономом максимального степеня.
3) Повторити попередній крок P разів, кожного разу змінюючи значення якогось випадкового біту із фіксованого набору.
4) Якщо моном максимального степеня не з'являється жодного разу або з'являється P разів, то такий шифр можна відрізнити від випадкового.

У роботі [2] було запропоновано для шифру з l раундами ініціалізації розглядати не просто наявність моному максимального степеня у АНФ булевої функції після ініціалізації шифру, а l різних булевих функцій і наявність моному максимального степеня у АНФ кожної з них. Таким чином можна з'ясувати, чи є кількість раундів ініціалізації шифру достатньою для встановлення складних функціональних зв'язків

між бітами секретного ключа та бітами гами. Послідовність із l коефіцієнтів мономів максимального степеня називається *MDM-сигнатурою*. Для ідеального шифру MDM-сигнатура повинна виглядати як випадкова послідовність.

Основна проблема використання такого алгоритму полягає у тому, що вибір різних множин S дає різні результати, а S, в свою чергу, можна обрати багатьма способами, які також неможливо перебрати на сучасних обчислювальних машинах. Нас цікавить максимальна кількість невипадкових значень у MDM-сигнатурі.

Жадібний алгоритм вибору підмножини бітів

Для вирішення проблеми вибору множини S у роботі [2] було запропоновано жадібний алгоритм вибору підмножини бітів, який додає до множини S ті біти, які дають найбільшу кількість початкових нулів у MDM-сигнатурі.

Але цей жадібний алгоритм має таку ж проблему, як і всі жадібні алгоритми, а саме – може потрапити у локальний максимум замість глобального. Саме тому у роботі [4] було запропоновано покращений жадібний алгоритм, який працює так само, як і звичайний жадібний алгоритм, але завжди потрапляє до глобального максимуму.

Покращений жадібний алгоритм ґрунтується на думці, яка полягає в тому, щоб розглядати багато можливих «гілок». На кожному кроці замість того, щоб розглядати лише один найкращий біт, ми розглядаємо кілька кращих бітів та зберігаємо їх для можливого використання у подальшому. Основна ідея такої реалізації полягає в тому, що другий найкращий кандидат в один крок із досить високою ймовірністю може бути кращим на наступній ітерації алгоритму.

Покращений жадібний алгоритм приймає на вхід такі параметри: секретний ключ K, вектор ініціалізації IV, потужність вихідної множини m та три вектор-параметри $k,\ \alpha,\ n$. На виході отримуємо множину бітів S потужності m, яка дає максимальну кількість початкових нулів у MDM-сигнатурі. Алгоритм можна описати такими логічними кроками.

1) Розглядаємо набір бітів-кандидатів із попередньої ітерації або з оптимального стартового набору.
2) Для кожного кандидата у списку ми додаємо n_i найкращих нових бітів k_i та зберігаємо їх у новому списку. Зауважимо, що тепер у нас є один такий новий список для кожного кандидата в оригінальному списку.
3) Об'єднуємо всі списки, сортуючи за кількістю нулів у MDM-сигнатурі.
4) Нарешті, зменшуємо розмір цього списку з урахуванням коефіцієнта α_i $(0 < \alpha_i \leq 1)$.
5) Повторюємо усі кроки, починаючи з кроку 1, доти, доки не буде знайдено множину бітів необхідної потужності.

Автори алгоритму у роботі [4] зазначають, що звичайний жадібний алгоритм вибору бітів є особливим випадком даного узагальненого алгоритму, який відповідає конкретним значенням векторів-параметрів $\alpha = [1.0, 1.0, ...], k = [1, 1, ...], n = [n, n, ...]$.

Також треба зазначити те, що «жадібність» алгоритму можна досить гнучко змінювати завдяки зміні значень вектор-параметрів k, α, n. Це дозволяє точно налагодити алгоритм для певного шифру. Тому покращений жадібний алгоритм є досить потужним інструментом для аналізу потокових алгоритмів шифрування.

Алгоритм шифрування «Струмок» виконує ініціалізацію за 32 раунди [1], тобто ми досліджуємо $l = 32$ різні булеві функції, і MDM-сигнатура – це послідовність із 32 елементів.

Для того, щоб провести таку атаку на шифр «Струмок», шифр було модернізовано так, щоб він породжував гаму під час ініціалізації, звідки ми й беремо перший біт.

Висновки

У результаті застосування покращеного жадібного алгоритму вибору бітів до алгоритму шифрування «Струмок» планується отримати MDM-сигнатуру із максимальною можливою кількістю початкових нулів. Ця MDM-сигнатура може бути використана для перевірки достатності кількості раундів ініціалізації шифру «Струмок».

Перелік використаних джерел

1. О.О. Кузнецов, І.Д. Горбенко, Ю.І. Горбенко, А.М. Олексійчук. Математична структура потокового шифру Струмок. Харківський національний університет імені В.Н. Каразіна. — 2018.
2. Stankovski, P. (2010). Greedy distinguishers and nonrandomness detectors. In INDOCRYPT 2010, pages 210–226. Springer.
3. Englund, H., Johansson, T., and Sönmez Turan, M. (2007). «A framework for chosen IV statistical analysis of stream ciphers. » In Progress in Cryptology – INDOCRYPT 2007: 8th International Conference on Cryptology in India, Chennai, India, December 9-13, 2007. Proceedings, pages 268–281. Springer.
4. Karlsson, Linus; Hell, Martin; Stankovski, Paul «Improved Greedy Nonrandomness Detectors for Stream Ciphers» Published in: Proceedings of the 3rd International Conference on Information Systems Security and Privacy, 2017.
5. S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the key scheduling algorithm of RC4. Lecture Notes in Computer Science, 2259:1–24, 2001.