УДК 003.26.09:004.032.24-004.272.3

А. М. Луцків, І. А. Вітрук, Н. В. Загородна
Тернопільський національний технічний університет імені Івана Пулюя,
кафедра комп'ютерних систем та мереж

ВИСОКОПРОДУКТИВНА ІНФОРМАЦІЙНА СИСТЕМА АЛГЕБРАЇЧНОГО КРИПТОАНАЛІЗУ ПОТОКОВИХ ШИФРІВ

Розроблено високопродуктивну інформаційну систему для алгебраїчного криптоаналізу потокових шифрів, запропоновано підхід до декомпозиції обчислювальної задачі й проведено обчислювальний експеримент. Показано, що час розв'язання системи алгебраїчних рівнянь можна зменшити за допомогою технологій паралельного програмування, зокрема OpenMP and MPI.

Ключові слова: алгебраїчний криптоаналіз, високопродуктивні обчислення, Cryptominisat, потокові шифри.

HIGH-PERFORMANCE INFORMATION SYSTEM FOR STREAM CIPHER CRYPTANALYSIS BY ALGEBRAIC METHOD

Article deals with development of high-performance information system for stream cipher cryptanalysis by algebraic method. Decomposition of computational problem was suggested and computational experiment was carried out. Execution time of SAT solving can be decreased by using parallel programming technologies especially OpenMP and MPI.

Key words: algebraic cryptanalysis, HPC, Cryptominisat, stream ciphers.

Вступ

В ході верифікації криптостійкості алгоритмів шифрування доцільним є використання усіх теоретично та практично можливих й відомих тепер алгоритмів криптоаналізу. Порівняно новим є метод алгебраїчного криптоаналізу [1–3], який застосовують для тестування багатьох блокових та потокових шифрів. Особливістю алгебраїчного криптоаналізу є опис криптографічного алгоритму та його параметрів системою алгебраїчних рівнянь великої розмірності. А криптоаналіз полягає у розв'язанні такої системи рівнянь. Доцільно мінімізувати час здійснення криптоаналізу, розпаралеливши його, для чого потрібно модифікувати наявне алгоритмічне та програмне забезпечення. Водночас, сукупність програмного, в основу якого покладено алгоритмічне та апаратне забезпечення, формує порівняно складну інформаційну систему з взаємопов'язаними компонентами. Й створення ефективної інформаційної системи криптоаналізу передбачає урахування особливостей усіх її складових: алгоритмічного, програмного та апаратного.

Постановка проблеми

У цій роботі автори, на основі аналізу алгоритмічного та програмного забезпечення алгебраїчного криптоаналізу, пропонують підхід до декомпозиції обчислювальної задачі. Здійснено обчислювальний експеримент з метою перевірки ефективності запропонованого підходу. Для дослідження використано відкрите програмне забезпечення та відомі алгоритми шифрування. Апаратне забезпечення системи паралельного та розподіленого опрацювання даних формується за принципом доступності та універсальності її компонент [4].

Аналіз останніх досліджень та публікацій

Коротко проаналізуємо останні дослідження та розробки у цій предметній області: математичне, алгоритмічне та програмне забезпечення алгебраїчного криптоаналізу потокових шифрів. Варто зазначити, що вищезгадані шифри використовують аналогічні принципи, а відповідно аналогічними є й методи їх криптоаналізу.

Одним із шифрів, які аналізують у ході дослідження, є Hitag2. Це синхронний потоковий шифр на основі лінійних регістрів зсуву зі зворотними зв'язками (РЗЛЗЗ) і нелінійної об'єднувальної функції. Використовується в охоронних системах, а саме RfID-ключах до електронних автомобільних замків (BMW, AUDI, Alfa Romeo, Porsche, Bentley, Volkswagen, Peugeot, Renault, Citroen, Iveco, Ford, GM, Volvo та інших) й випускається компанією NXP Semiconductors [5].

Hitag2 складається з генератора на основі 48-бітного РЗЛЗЗ і нелінійної функції з 20 входами, котра формує один вихідний біт за один такт. Інші компоненти використовуються тільки на етапі ініціалізації. Hitag2 працює з 48-бітним секретним ключем, який доступний для пристроїв читання, запису і транспондера, 32-бітним серійним номером тегу і 32-бітним вектором ініціалізації IV.

Нелінійна функція об'єднання з 20 входами шифру Hitag2 має високу криптостійкість, проте результат цієї нелінійної функції записується в РЗЛЗЗ тільки під час фази ініціалізації. Відсутність нелінійних вхідних даних протягом фази генерування ключового потоку зменшує загальну складність шифру. Цю властивість використано в алгебраїчних криптоатаках [3].

Ще одним прикладом потокового симетричного алгоритму шифрування є A5/1, який використовується у стільникових системах зв'язку стандарту GSM для шифрування голосових даних. Для шифрування даних (технологія GPRS) використовується криптоалгоритм GEA1, принципи роботи якого аналогічні до A5/1. A5/1 також базується на РЗЛЗЗ: L1, L2 і L3 з довжинами регістрів 19, 22 і 23 біт. Аналіз його лінійних булевих функцій також вказує на вразливість до алгебраїчних атак.

Завдання алгебраїчного криптоаналізу потокового шифру полягає у знаходженні початкового стану, заданого деякими ключовими потоками бітів (k_0,..., k_{n-1}) з деякими m послідовними бітами гами b_0... b_{m-1}, розв'язанням систем багатовимірних рівнянь (так звані "швидкі алгебраїчні атаки" [3] передбачають наявність послідовних бітів). Є різні тісно пов'язані підходи до алгебраїчної атаки на потокові шифри, що різняться переважно використовуваними типами рівнянь, які й визначають методи їх розв'язання. Так, атаки на основі прямого рівняння застосовують тільки для станів без об'єднувача. Нехай (k_0,..., k_{n-1}) є початковим станом, тоді на виході шифру (тобто як ключовий потік) отримаємо таку систему рівнянь:

$$\begin{cases} b_0 = f(k_0,...,k_{n-1}) \\ b_1 = f(L(k_0,...,k_{n-1})) \\ b_1 = f(L^2(k_0,...,k_{n-1})) \end{cases} \quad (1)$$

Такі рівняння спочатку множать на певні багатовимірні поліноми (багаточлени) для зменшення їх степеня, після чого розв'язують.

Також одним із видів рівнянь, які генеруються із зашифрованого тексту відповідного алгоритму шифрування, є булеві рівняння. А задача, що зводиться до розв'язання таких булевих рівнянь, називається задачею здійсненності булевих функцій (з англ. *satisfiability*/SAT).

Булева функція (рівняння) називається здійсненною, якщо для всіх $x_i, i = 1..N$, де N – кількість змінних булевої функції, можна представити такі логічні значення "істинності" або "хибності", щоб загалом булева функція набувала значення "істинної". Комп'ютерна програма, що розв'язує задачі здійсненності булевих функцій, називається "SAT-solver". Останнім часом "SAT solver'и" знаходять усе більше застосувань у галузі криптоаналізу. У роботі [3] показано, що такий вид програм може успішно застосовуватись для алгебраїчного криптоаналізу блокових та потокових шифрів. Вхідними даними відповідних програм є задачі, описані в кон'юнктивній нормальній формі (КНФ, тобто КНФ-файли). На виході таких програм отримують набір розв'язків рівняння, що дає змогу встановити ключ шифрування для початкового зашифрованого повідомлення.

Серед багатьох засобів автоматизованого роз’язання рівнянь у КНФ одним із найвідоміших є Cryptominisat [7]. Cryptominisat ґрунтується на використанні алгоритму Девіса–Патнема–Логемана–Лавленда (DPLL) [8], який є повним алгоритмом пошуку з поверненням для визначення здійсненності булевих функцій, які записані у кон’юнктивній нормальній формі. Проте у цієї програми є недоліки. Так, час її виконання іноді є порівняно великим, тому доцільно вдосконалювати запропоноване програмне забезпечення на рівні його алгоритмів. Варто зазначити, що таке програмне забезпечення поширюється під відкритою ліцензією LGPL, а це дає змогу здійснювати його вдосконалення й безперешкодно використовувати в ході досліджень.

Також доволі складним завданням у цьому різновиді алгебраїчного криптоаналізу є формування КНФ-файлів, тобто конвертування опису шифру до його відповідного представлення в КНФ-формі. Серед автоматизованих засобів для розв'язання цієї задачі найвідоміші такі: Python-модуль, розроблений Martin Albrecht для математичної платформи sage [9], Logic2CNF – розроблений Edd Barrett [10], і STP (Simple Theorem Prover), автором якого є Ganesh [11]. Ці інструменти пропонують широкий набір можливостей і можуть бути використані на різних рівнях абстракції. Наприклад, Logic2CNF лише перетворює опис шифру з алгебраїчної нормальної форми (АНФ) в КНФ, але не може генерувати АНФ з поданого опису шифру. STP може сформувати повний опис шифру, але не зберігає його в АНФ, опускаючи можливість оптимізації на рівні абстракції. Sage-модуль лише перетворює вже описаний раніше шифр на КНФ опис, але не може використати інструменти, призначені для опрацювання задачі в АНФ. Більшість із згаданих програм та інструментів реалізують власні способи представлення шифру в КНФ. В аспекті зазначених недоліків виділимо програму Grain of Salt [12] – інструмент, що генерує оптимізовані КНФ-файли, базуючись на описі відповідного потокового шифру. Функціональні можливості Grain of Salt дають змогу генерувати КНФ для будь-яких потокових шифрів на основі регістрів зсуву (Grain, Trivium, BiviumB, HiTag2, Crypto1 та інших). Ця програма має широкий набір опцій, зокрема таблиці мінімізації Карно, дає змогу здійснювати розширену обробку одночленів, створювати статистичні дані й здійснювати візуалізацію процесу обчислення. Варто зазначити, що всі вищезгадані пакети програм є відкритим програмним забезпеченням, що дає змогу здійснювати їх модифікацію у власних цілях. Очевидно, що в ході цього криптоаналітичного дослідження доцільним є представлення шифротексту у вигляді системи рівнянь у КНФ й використання програми Cryptominisat та Grain Of Salt.

Апаратна складова інформаційної криптоаналітичної системи основана на доступному апаратному забезпеченні [4].

Формулювання цілі статті

Отже, проаналізовано програмні засоби алгебраїчного криптоаналізу потокових шифрів й відповідні їм методи, окреслено основні проблеми наявних компонент інформаційних систем криптоаналізу. Основні завдання під час створення криптоаналітичних систем такі:

- модифікація відомих криптоаналітичних методів алгебраїчного криптоаналізу потокових шифрів шляхом їх розпаралелення та векторизації;
- розроблення та модифікація наявного програмного забезпечення з урахуванням можливості його паралельного, векторного та гібридного виконання на доступному апаратному забезпеченні;
- проведення обчислювального експерименту на основі запропонованого криптоаналітичного програмного забезпечення, з метою верифікації запропонованих підходів, які покладено в основу роботи криптоаналітичної системи.

Для вирішення вказаних завдань необхідно розв’язати низку задач:

1. Здійснити вибір та обґрунтування методу алгебраїчного криптоаналізу, який є основою роботи криптоаналітичної системи.
2. Оскільки досліджувані методи послідовні – розробити методи їх декомпозиції та розпаралелення.
3. Реалізувати запропоновані алгоритми та методи у вигляді програм, з модифікацією (використанням) існуючих та створенням власних програмних компонент.

4. Забезпечити узгодженість роботи апаратних та програмних компонент інформаційної системи.

5. Провести обчислювальний експеримент та здійснити аналіз отриманих результатів

У розв’язанні цієї задачі можна виділити такі етапи:

1) попереднє опрацювання шифротексту та його декомпозиція за даними;
2) генерування систем рівнянь у кон’юнктивній нормальній формі (КНФ);
3) розподіл між обчислювальними вузлами;
4) розв'язання систем рівнянь та інтерпретація результатів.

Нижче розглянемо ці етапи детальніше.

Виклад основного матеріалу

На основі аналізу предметної області, з метою мінімізації часу виконання обчислювальної задачі, запропоновано підхід до її декомпозиції, що розглянемо на прикладі алгоритму шифрування Hitag2. Цей шифр є синхронним, а тому формування вмісту регістру зсуву залежить лише від корисної інформації та від початкового 48-бітного ключа, й не залежить від попередньо сформованих шифротекстів. Тобто відсутня залежність між даними, а отже, доцільне використання декомпозиції за даними.

До прикладу, на вході криптоаналітичної системи є шифротекст довжиною n бітів. Кожний наступний біт шифротексту утворений незалежно від попередніх бітів. Вихідний біт формується лише на основі операції XOR між корисними вихідними даними та вмістом регістра зсуву, а також функціями фільтрації й, отже, рекурсивні залежності й зв’язки між різними фрагментами даних відсутні.

Отже, вихідну послідовність шифротексту з l бітів можна поділити на K фрагментів, де K – кількість обчислювальних вузлів, кожен з яких розв’язуватиме задачу для l/K біт шифрованого тексту. Проте час розв’язання задачі програмою Cryptominisat, що використовує алгоритм DPLL, нелінійно залежить від довжини досліджуваного шифротексту в бітах, адже наперед неможливо передбачити, на якій вітці пошукового дерева може завершитись пошук. Для встановлення відповідності між довжиною фрагмента шифротексту та часом його опрацювання програмою використано емпіричний метод дослідження.

Експерименти проводились на чотиривузловій комп'ютерній системі з такими характеристиками: операційна система – Linux Ubuntu 12.10, процесор – Intel Core i5 3317u, 2 Гб оперативної пам'яті. Дані в таблиці підтверджують, що кількість змінних і кількість рівнянь у згенерованих КНФ-файлах збільшується нелінійно. На рисунку графічно представлено співвідношення часу розв’язання задачі та довжини зашифрованого фрагмента тексту, який опрацьовується у Cryptominisat.

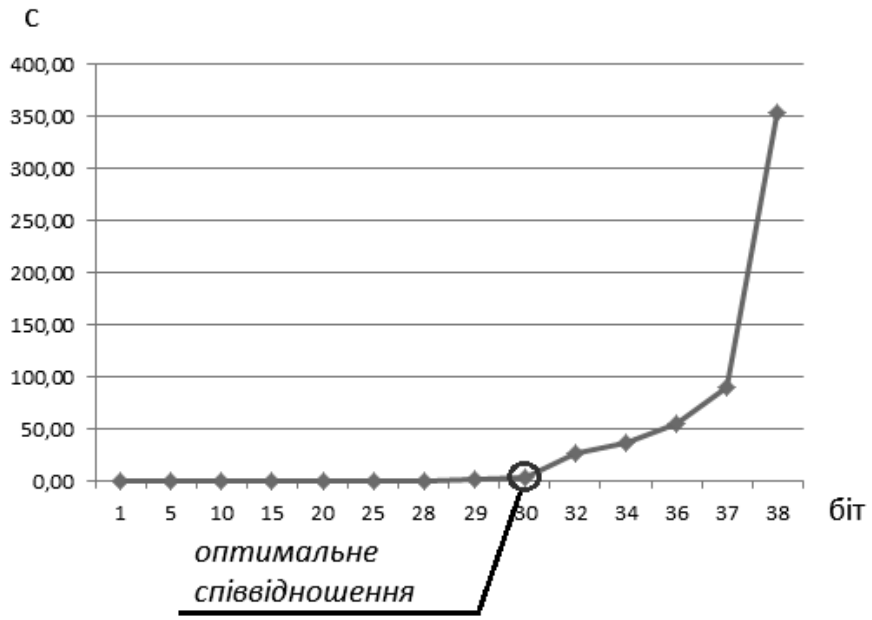
Розпаралелення на рівні програмного забезпечення здійснено засобами технологій OpenMP (система зі спільною пам’яттю) та MPI (система з розподіленою пам’яттю). Тобто багатоядерні вузли об’єднані в обчислювальний кластер [4].

Проаналізувавши графік, представлений на рисунку, бачимо, що час опрацювання фрагмента довжиною до 30 бітів близький до нуля. Після збільшення до 32 бітів час зростає до 25 секунд, а для 38 бітів становить приблизно 6 хвилин. Тому для декомпозиції вхідних даних для розв’язання програмою Cryptominisat вибрано оптимальне співвідношення між довжиною і часом – 30 бітів. Тобто, маючи K вузлів кластера, шифротекст завдовжки l бітів, потрібно розділити на фрагменти по 30 бітів (таких фрагментів буде $l/30$), а потім розподілити рівномірно між кількістю вузлів. Кожен з K вузлів виконуватиме по $\frac{l}{30 \cdot K}$ – фрагментів завдань, довжина яких по 30 бітів. Для попередньої підготовки даних використовуємо програмний пакет Grain of Salt, яким генеруємо необхідну кількість обчислювальних задач (в цьому випадку $l/30$) у КНФ, тобто проводимо декомпозицію за даними, а також здійснюємо мінімізацію розмірності систем рівнянь (на основі карт Карно). Після

цього кожен вузол обчислювальної системи виконує над даними однакові дії, тобто засобами програми Cryptominisat розв'язує систему рівнянь згідно зі своїм фрагментом. Кожен вузол міститиме той фрагмент потоку гами регістра зсуву, що відповідає за шифрування відповідного фрагмента шифротексту. Коли вузли закінчать обчислення, вони відправлять результуючі дані до головного вузла, який збере ці дані і, з'єднавши їх, сформує загальну гаму, яка використовувалась у регістрі зсуву в процесі шифрування. У результаті наведеної схеми криптоаналізу, на основі шифротексту й повної гами регістру зсуву, можна отримати відкритий текст частинами по 80 бітів інформації, перші 32 з яких будуть корисними даними, а 48 інших – ключ.

Результати обчислювального експерименту

Довжина фрагмента шифротексту, біт	Кількість змінних КНФ-файла	Кількість рівнянь	Обчислювальний експеримент, с			Середній час, с
			№ 1	№ 2	№ 3	
1	91	419	0,01	0,01	0,01	0,01
5	272	2677	0,06	0,06	0,07	0,06
10	497	5727	0,15	0,16	0,17	0,16
15	707	8732	0,31	0,33	0,35	0,33
20	912	11722	0,42	0,41	0,43	0,42
25	1117	14712	0,57	0,65	0,59	0,60
28	1240	16506	0,90	0,94	0,92	0,92
29	1281	17104	1,20	1,30	1,20	1,23
30	1322	17702	2,10	2,60	2,20	2,30
32	1404	18898	27,00	26,90	27,40	27,10
34	1486	20094	36,90	35,70	36,00	36,20
36	1568	21290	55,20	54,80	55,00	55,00
37	1609	21885	86,00	94,00	89,00	89,67
38	1650	22486	361,00	346,00	353,00	353,33



Залежність довжини фрагмента від часу розв'язання

Наприклад, пристроєм шифрується 30000 бітів (3,75 Kb) корисної інформації й генерується 30000 бітів зашифрованого тексту. Використовуючи запропоновану криптоаналітичну інформаційну систему та вищезгадані методи, розшифруємо шифротекст й відновимо корисну інформацію та ключ шифрування, який використовується у HITAG2.

Оскільки отримані в ході дослідження результати вказують на ефективну довжину фрагмента шифротексту в 30 бітів, то відповідно до цього розмір КНФ-файлів, які формуватимуться для їх подальшого опрацювання обчислювальними вузлами кластера, становить $K_N > \frac{l}{k * 30}$, де $K_N = 4$ – максимальна кількість обчислювальних вузлів; $l = 30000$ – довжина шифротексту; k – кількість КНФ-файлів для розв'язання кожним вузлом. Отже, $k = \frac{30000}{4 * 30} = 120$.

Отже, кожен вузол опрацьовує 120 КНФ файлів. Кожен вузол використовуватиме локальну пам’ять і процесорні ядра для обчислень (використано технологію OpenMP). У результаті буде отримано розв'язання завдовжки 1322 бітів. Для 30 бітів шифротексту потрібно 1440 змінних, які формують систему рівнянь у АНФ. Загальна довжина розв'язання, яка отримується з усіх вузлів, становить 1322000 біти й дає змогу відновити 32 біти зашифрованих даних та 48 бітів ключа.

Висновки

Запропоновано підходи до декомпозиції обчислювальної задачі алгебраїчного криптоаналізу потокових шифрів, обґрунтовано вибір програмного забезпечення та створено власні компоненти програмної складової інформаційної системи криптоаналізу потокових шифрів. Здійснено обчислювальний експеримент, який показав ефективність запропонованого підходу з оптимальною довжиною фрагмента зашифрованого тексту 30 бітів.

У подальших дослідженнях планується досягти більшої ефективності, використавши апаратні засоби GPGPU.

1. Кравець О. Підвищення ефективності криптоаналізу сучасних потокових шифрів / О. Кравець, С. А. Лупенко, А. М. Луцків // Вісник національного університету “Львівська політехніка”. – Львів: Видавництво Львівської політехніки, 2012. – № 741. – С. 240–245. 2. Nicolas Courtois: Higher Order Correlation Attacks, XL algorithm and Cryptanalysis of Toyocrypt, ICISC 2002, LNCS 2587, Springer. [Електронний ресурс]: Access mode: URL: http://eprint.iacr.org/2002/087/. 3. Nicolas Courtois and Willi Meier: Fast Algebraic Attacks on Stream Ciphers with Linear Feedback, Eurocrypt 2003, Warsaw, Poland, LNCS 2656, pp. 345-359, Springer. An [Електронний ресурс]: Access mode: URL: http://www.minrank.org/toyolili.pdf. 4. Загородна Н. В. Обґрунтування вибору доступних програмно-апаратних засобів високопродуктивних обчислювальних систем для задач криптоаналізу / Н. В. Загородна, С. А. Лупенко, А. М. Луцків // Електроніка та системи управління. – 2011. – № 1(27). – К.: НАУ, 2011. – С.42–50. 5. Security Transponder (HITAG2) Product Specification. PCF7936AS. NXP Semiconductors [Електронний ресурс]: Режим доступу: URL: http://www.mouser.com/catalog/specsheets/PCF7936AS__3851__C,1.pdf. 6. Petr Stembera. Cryptoanalysis of hitag2 cipher / Czech Technical University in Prague, 2011 – 61 p. 7. Mate Soos – CryptoMiniSat3 [Електронний ресурс]: Access mode: URL: http://www.msoos.org/cryptominisat3/. 8. Nieuwenhuis, Robert; Oliveras, Albert & Tinelly, Cesar (2004), “Abstract DPLL and Abstract DPLL Modulo Theories”, Logic for Programming, Artificial Intelligence, and Reasoning, LPAR 2004, Proceedings: 36–50. 9. The SAGE Group: SAGE mathematics software (2008) [Електронний ресурс]. – Режим доступу: URL: http://www.sagemath.org. – Назва з екрана. 10. Barrett E.: Logic2CNF logic solver and converter (March 2010) [Електронний ресурс]. – Режим доступу: URL: http://projects.cs.kent.ac.uk/projects/logic2cnf/ – Назва з екрана. 11. Ganesh V., Dill, D.L.: A decision procedure for bit-vectors and arrays. In Damm, W., Hermanns, H., eds.: CAV. Volume 4590 of Lecture Notes in Computer Science., Springer (2007) 519-531. 12. Soos M. Grain of Salt - An Automated Way to Test Stream Ciphers through SAT Solvers” [Електронний ресурс]: Access mode: URL: http://www.msoos.org/grain-of-salt