УДК 004.056

М.Карпінський[1], докт.техн.наук; Л.Коркішко[2];
Т.Коркішко[3], канд.техн,наук
[1]*Університет в Бєльську-Бялей, м. Бєльсько-Бяла, Республіка Польща*
[2]*Тернопільська академія народного господарства*
[3]*Інститут передових технологій Самсунг, м. Сеул, Південна Корея*

ІНЖЕНЕРНО-КРИПТОГРАФІЧНА АТАКА ЗА АНАЛІЗОМ СПОЖИВАНОЇ ПОТУЖНОСТІ НА ПРОГРАМНО-АПАРАТНІ РЕАЛІЗАЦІЇ КРИПТОГРАФІЧНОГО ПЕРЕТВОРЕННЯ ЗА ЧИННИМ СТАНДАРТОМ

Запропоновано методику проведення атаки на комп'ютерні реалізації алгоритму криптографічного перетворення за чинним в Україні стандартом колишнього СРСР і висвітлено особливості її застосування. Проведено оцінку часової складності проведення цієї атаки на комп'ютерні реалізації алгоритму криптографічного перетворення. Доведено, що для визначення ключа шифрування при цьому витрачається значно менший час порівняно з прямим перебором з відомими вузлами таблиці підстановки чи аналізом, який базується на особливостях процедури розпису ключа шифрування.

Вступ

Із розширенням областей застосування електронного обміну інформацією загострюється проблема неавторизованого доступу до даних, які передаються, зберігаються чи обробляються. Одним із можливих варіантів вирішення цієї проблеми є застосування до даних, перед їх відправленням, криптографічних перетворень, наприклад зашифрування із використанням невідомих даних – таємного ключа. При цьому можливість доступу до даних, які піддавалися зашифруванню без початкових відомостей про ключ зашифрування, виникає при компрометації алгоритму шифрування (успішного його математичного криптографічного аналізу). У сучасних комп'ютерних системах, які реалізують криптографічні перетворення для забезпечення захисту даних, використовуються достатньо стійкі до математичного криптографічного аналізу алгоритми криптографічних перетворень. Тому для визначення ключа шифрування і подальшого доступу до даних часто використовуються спеціальні види криптоаналізу. Результати робіт [1–8] дозволяють розглядати аналіз такого роду як "інженерно-криптографічний", оскільки проводиться аналізування реалізацій алгоритмів криптографічних перетворень, тобто інженерних виробів, а не абстрактних математичних об'єктів, як у випадку класичного криптографічного аналізу. При цьому для отримання інформації про особливості обчислювального процесу з використанням таємної інформації використовується інформація, отримана з побічних каналів її витоку. Основними побічними каналами є час виконання криптографічних операцій, споживана потужність пристрою, електромагнітне випромінювання пристрою [1 – 8]. Існування цих каналів пояснюється відмінністю фізичних процесів, які відбуваються у пристрої при виконанні обчислень над різними даними. Тому інженерно-криптографічний аналіз передбачає використання інформації, яка отримується шляхом спостереження за роботою пристроїв, які реалізують криптографічні перетворення. Цей аналіз часто називають інженерно-криптографічними атаками. Виходячи з цього, актуальною задачею при розробці комп'ютерних пристроїв для реалізації криптографічних перетворень є мінімізація інформації, яка доступна за побічними каналами її витоку.

Відомі роботи з проведення інженерно-криптографічного аналізу комп'ютерних реалізацій алгоритмів криптографічних перетворень, наприклад, DES [3, 4, 6], RSA [2, 5], AES [6] тощо. Однак проведення згаданих атак на реалізації алгоритму криптографічного перетворення, згідно з нормативним документом колишнього СРСР

ГОСТ 28147-89 [9], який прийнято за стандарт в Україні, недостатньо висвітлено у літературі.

Базою для проведення цього аналізу є інженерно-криптографічні атаки з використанням статистичних моделей складових операцій криптографічних перетворень та моделі витоку інформації з комп'ютерних засобів. Алгоритм криптографічного перетворення за стандартом ГОСТ 28147-89 використовує операції логічного додавання за модулями 2 та 2^{32}, циклічного зсуву, операцію підстановки. Оскільки статистична модель для побітової операції логічного додавання за модулем 2 є достатньо добре описаною та дослідженою [7, 8], а статистична модель для операції логічного додавання за модулем 2^N висвітлена у [10], дана робота присвячена актуальній задачі створення методики проведення інженерно-криптографічної атаки з використанням інформації про споживану потужність програмованого комп'ютерного пристрою, який реалізує алгоритм криптографічного перетворення згідно зі стандартом ГОСТ 28147-89. Для оцінки ефективності і перспективності практичного використання цієї атаки у статті проведено оцінку її часової складності [11].

1. Алгоритм криптографічного перетворення згідно зі стандартом ГОСТ 28147-89

Алгоритм криптографічного перетворення у стандарті ГОСТ 28147-89 визначає правила шифрування даних та обчислення імітовставки (спеціального блоку, який обчислюється з послідовності вхідних даних та ключа шифрування) [9]. Згідно з алгоритмом проводиться перетворення 64-бітових блоків даних із використанням ключа шифрування. У стандарті ГОСТ 28147-89 прийняті такі режими обробки даних: режим простої заміни, режим гамування, режим гамування зі зворотним зв'язком і режим вироблення імітовставки. Нижче розглянуті особливості режиму простої заміни, який є основою для реалізації решти режимів.

Загалом обробка даних виконується за 32 цикли (раунди). З вхідного ключа формуються тридцять два 32-бітові ключі для кожного циклу окремо. Алгоритми зашифрування та розшифрування відповідають схемі Фейстеля [12], у якій раундова функція задана послідовністю логічного додавання за модулем 2^{32}, табличними підстановками, що виконуються над чотирибітовими підблоками, та операцією циклічного зсуву ліворуч на 11 бітів, що виконується над 32-бітовим підблоком. Усі раунди є функціонально еквівалентними.

Головним елементом алгоритму є цикл, який перетворює 64-бітовий вхідний блок у 64-бітовий вихідний блок даних з використанням 32-бітового циклового ключа.

Приймемо, що X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7 – 32-розрядні циклові ключі, N_1, N_2 – проміжні 32-розрядні бітові вектори. У випадку логічного додавання та циклічного зсуву старшими вважають розряди з більшими номерами.

Операція підстановки K – це підстановка вхідного 32-бітового блоку даних згідно з таблицею підстановки, що задана вузлами підстановки K_0, K_1, K_2, K_3, K_4, K_5, K_6, K_7 розміром 64 біти кожний. 32-розрядний вектор розбивається на вісім 4-розрядних векторів, які надходять послідовно і кожний з яких перетворюється в 4-розрядний вектор відповідним вузлом підстановки. Вузол підстановки є таблицею з шістнадцяти стрічок, які містять по чотири біти заповнення в стрічці. Вхідний вектор визначає адресу стрічки в таблиці, заповнення цієї стрічки є вихідним вектором. Далі 4-розрядні вихідні вектори послідовно об'єднуються у 32-розрядний вектор. Алгоритм передбачає використання таких операцій: "$\otimes$" – логічне додавання за модулем 2^{32}, "$\oplus$" – логічне додавання за модулем 2, "R" – циклічний зсув 32-бітового вектора в бік старших розрядів на 11 бітів, тобто

$$R(r_{32},r_{31},r_{30},r_{29},r_{28},r_{27},r_{26},r_{25},r_{24},r_{23},r_{22},r_{21},r_{20},\ldots,r_2,r_1) =$$
$$= (r_{21},r_{20},\ \ldots,r_2,r_1,r_{32},r_{31},\ r_{30},r_{29},r_{28},r_{27},r_{26},r_{25},r_{24},r_{23},r_{22}).$$

Ключ шифрування (W_1, W_2, ..., W_{256}), $W_q \in \{0, 1\}$, $q = 1, \ldots, 256$ записують послідовно у 32-розрядні бітові вектори ключа $X_0, ..., X_7$. Вміст восьми 32-розрядних векторів $X_0, X_1, ..., X_7$ має такий вигляд:

$$X_0 = (W_{32}, W_{31}, \ldots, W_2, W_1);$$
$$X_1 = (W_{64}, W_{63}, \ldots, W_{34}, W_{33});$$
$$\ldots$$
$$X_7 = (W_{256}, W_{255}, \ldots, W_{226}, W_{225}).$$

Відкриті дані, що підлягають зашифруванню, розбивають на блоки по 64 біти. Блок даних розбивається на дві частини і присвоюється векторам N_1 (права частина) і N_2 (ліва частина).

Алгоритм зашифрування 64-розрядного блоку відкритих даних складається з 32-ох циклів. У першому циклі початкове значення вектора N_1 додається за модулем 2^{32} до значення вектора X_0, при цьому значення вектора N_1 зберігається. Результат додавання перетворюється за таблицею підстановки K, і отриманий вектор циклічно зсувається на одинадцять бітів у бік старших розрядів. Результат зсуву додається за модулем 2 до 32-розрядного значення вектора N_2. Отриманий результат присвоюється N_1, при цьому попереднє значення N_1 присвоюється N_2. Перший цикл закінчується.

Наступні цикли виконуються аналогічно. У другому циклі використовується цикловий ключ X_1, в третьому циклі – цикловий ключ X_2 і так далі. У восьмому циклі використовується цикловий ключ X_7. В циклах з дев'ятого по шістнадцятий, а також з сімнадцятого по двадцять четвертий циклові ключі використовуються у тому ж порядку: $X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7$. В останніх восьми циклах – з двадцять п'ятого по тридцять другий – порядок використання циклових ключів зворотний: $X_7, X_6, X_5, X_4, X_3, X_2, X_1, X_0$. Отже, у випадку зашифрування в 32 циклах порядок вибору циклових ключів виконується так:

$$X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7,$$
$$X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_7, X_6, X_5, X_4, X_3, X_2, X_1, X_0.$$

У тридцять другому циклі результат логічного додавання за модулем два присвоюється вектору N_2, а вектор N_1 зберігає попереднє значення. Отримані після тридцять другого циклу зашифрування значення векторів N_1 і N_2 є блоком зашифрованих даних. Схематичне зображення одного циклу алгоритму шифрування показано на рис. 1.

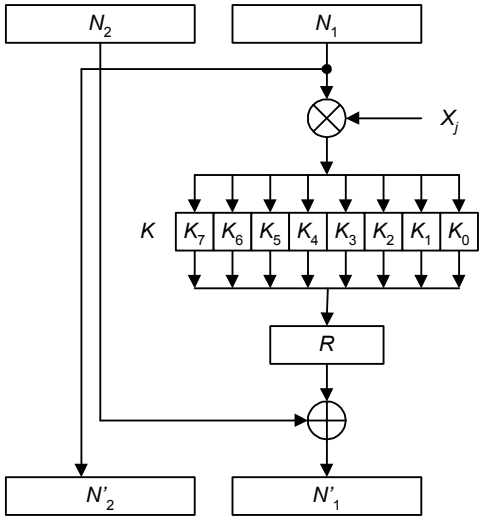

Рис. 1. Схема циклу алгоритму криптографічного перетворення за стандартом ГОСТ 28147-89:
N'_1, N'_2 – нові значення векторів N_1, N_2; X_j – цикловий ключ, $j = 0, \ldots, 7$

Алгоритм розшифрування в цілому аналогічний до алгоритму зашифрування даних. Різниця між алгоритмами зашифрування і розшифрування полягає у тому, що у

циклах розшифрування циклові ключі використовують у зворотному до зашифрування порядку.

2. Модель витоку інформації для інженерно-криптографічних атак

Для проведення інженерно-криптографічних атак приймемо, що:

- комп'ютерний пристрій, який реалізує алгоритм криптографічного перетворення, згідно зі стандартом ГОСТ 28147-89, дає можливість витоку інформації про Хемінгову вагу результатів виконання складових операцій;
- витік інформації відбувається через значення споживаного струму, а тому і через потужність, яку споживає пристрій;
- пристрій споживає більший струм при обробці даних з більшою Хемінговою вагою, залежність споживаного струму від Хемінгової ваги є лінійною.

Нехай споживання потужності у момент часу j подана у вигляді $P[j]$. Для моделювання каналу витоку інформації у сигналі $P[j]$ скористаємося лінійною залежністю, запропонованою у [8]:

$$P[j] = \varepsilon \cdot d[j] + L + n, \qquad (1)$$

де $d[j]$ репрезентує Хемінгову вагу результату, який отримується у момент часу j, ε – вклад у споживану потужність кожної одиниці Хемінгової ваги даних, L – постійна загальна споживана потужність, n – шум з нульовим середнім значенням.

3. Методика проведення інженерно-криптографічних атак на комп'ютерні реалізації алгоритму криптографічного перетворення за стандартом ГОСТ 28147-89

Відомі інженерно-криптографічні атаки на реалізації алгоритмів криптографічних перетворень мають на меті визначення ключа шифрування [1–8]. При цьому вважається, що крім ключа шифрування усі елементи та операції, які здійснюються за алгоритмом перетворення, є відомими. Цьому припущенню задовольняють зарубіжні алгоритми симетричного блокового шифрування, наприклад DES [13] і AES [14]. Однак алгоритм криптографічного перетворення, згідно зі стандартом ГОСТ 28147-89, передбачає використання значень вузлів таблиці підстановки, які, поряд із ключем шифрування, є таємними елементами [9]. Тому пряме використання відомих методик інженерно-криптографічних атак не є можливим, оскільки особливості проведення частини операцій алгоритму криптографічного перетворення не є відомими. А це, у свою чергу, унеможливлює побудову статистичної моделі для значень результатів циклових перетворень.

За даних передумов, у роботі пропонується будувати атаку на комп'ютерні реалізації алгоритму криптографічного перетворення ГОСТ 28147-89 із врахуванням статистичних властивостей відомих компонент перетворення, зокрема, операції логічного додавання за модулем 2^{32}, з подальшим визначенням таємних компонент – ключа шифрування і вузлів таблиці підстановки. Таким чином, атакування реалізації цього криптографічного перетворення буде здійснюватися пошарово:

- скориставшись статистичною моделлю суматора за модулем 2^{32} і методикою атакування реалізації цієї операції [10], визначимо цикловий ключ X_0;
- скориставшись отриманими відомостями про цикловий ключ X_0, статистичною моделлю суматора за модулем 2 і методикою атакування реалізації цієї операції [7, 8], визначимо вузли таблиці підстановки;
- скориставшись отриманими відомостями про цикловий ключ X_0, вузли таблиці підстановки, статистичною моделлю суматора за модулем 2^{32} і методикою атакування реалізації цієї операції [10], визначимо решту циклових ключів $X_1, X_2, ..., X_7$.

3.1. Методика визначення циклового ключа X_0

Особливістю обчислення розпису ключа (W_1, W_2, ..., W_{256}) алгоритму криптографічного перетворення за стандартом ГОСТ 28147-89 є використання процедури прямого формування циклових ключів X_0, X_1, X_2, X_3, X_4, X_5, X_6, X_7 з початкового ключа без виконання проміжних перетворень [15, 16]. Визначення циклових ключів шляхом повного перебору елементів простору ключа шифрування не є можливим внаслідок великої розмірності цього простору. Тому визначення ключа шифрування будемо проводити шляхом почергового визначення циклових ключів з використанням інформації, отриманої із аналізу споживаної потужності пристрою, який реалізує алгоритм криптографічного перетворення на підставі стандарту ГОСТ 28147-89.

Для визначення циклового ключа X_0 використаємо інформацію, отриману з аналізу споживаної потужності пристрою при виконанні першого циклу алгоритму криптографічного перетворення. Алгоритм циклового перетворення використовує операцію логічного додавання за модулем 2^{32}, статистична модель якої є частковим випадком статистичної моделі логічного додавання за модулем 2^N. Тому для визначення циклового ключа X_0 скористаємося статистичною моделлю суматора за модулем 2^N для $N = 32$ і методикою атакування реалізації цієї операції, розробленими в [10]. При цьому для збору даних про споживану потужність пристрою будемо використовувати лише часовий інтервал виконання першого циклу алгоритму:

```
Для і від 0 до 31 {
  Для b=0 до 1 {
     Обчислити усереднене значення сигналу споживаної потужності
   Ab[j] {
     Встановити і-й біт N2 рівним b;
     Якщо i>0, встановити біти N2 з номерами i-1,...,0 у нуль.
     Встановити решту бітів N2 з номерами 31,...,i+1 у випадкові
   значення;
     Зібрати дані про споживану потужність пристрою;
     }
  }
  Обчислити диференційний сигнал T[j] = A0[j] – A1[j];
  Якщо T[j] > 0, то і-й біт X0 є "1", якщо T[j] < 0 то і-й біт X0 є
"0";
 }
```

Наведений алгоритм дозволяє визначати усі біти циклового ключа X_0. При цьому постійні складові очікуваного значення Хемінгової ваги суми будуть змінюватися для кожного розряду. Аналогічного результату можна досягти, використавши модифікацію наведеного алгоритму, встановлюючи значення бітів з номерами меншими i в інвертовані значення вже визначених бітів X_0. При цьому очікуване значення Хемінгової ваги суми буде зростати для кожного невідомого розряду [10].

Далі розглянемо методику визначення невідомих значень вузлів таблиці підстановки.

3.2. Методика визначення вузлів таблиці підстановки

Для створення алгоритму виконання другого етапу атаки введемо позначення для першого циклу криптографічного перетворення:

- $X_0 = \left\{x_0^7 \| x_0^6 \| ... \| x_0^t \| ... \| x_0^1 \| x_0^0\right\}$, де x_0^t – 4-бітовий підблок X_0, $\|$ – позначення операції конкатенації підблоків;
- $N_1 = \left\{n_1^7 \| n_1^6 \| ... \| n_1^t \| ... \| n_1^1 \| n_1^0\right\}$, де n_1^t – 4-бітовий підблок N_1;

- $Y = \{y_7 \| y_6 \| ... \| y_t \| ... \| y_1 \| y_0\}$, де $Y = (X_0 + N_1) \bmod 2^{32}$, y_t – 4-бітовий підблок Y;
- $Q = \{q_7 \| q_6 \| ... \| q_t \| ... \| q_1 \| q_0\}$, де $q_t = K_t(y_t)$ – 4-бітовий підблок результату виконання операції підстановки над вхідним блоком y_t з використанням таблиці підстановки K_t, q_t приймає значення з набору $\{k_t^0, k_t^1, ..., k_t^h, ..., k_t^{14}, k_t^{15}\}$, k_t^h приймає значення 0, 1, ..., 15, а порядок розташування k_t^h не є відомим;
- $S = R(Q)$;
- $N'_1 = N_2 \oplus S$.

Визначення усіх невідомих значень вузлів підстановки K проведемо почергово для кожної таблиці підстановки K_t. Для визначення значень вузлів кожної з таблиць підстановки K_t зафіксуємо t і почергово визначимо вузли k_t^h для усіх h.

Оскільки значення X_0 є відомим і можна довільно встановлювати значення N_1, підберемо таке значення n_1^t, що $y_t = h = (x_0^t + n_1^t) \bmod 2^4$. Приймемо до уваги, що значення вузла заміни k_t^h не є відомим, а тому і не буде відомим значення q_t. Врахувавши, що Хемінгова вага результату S виконання операції R циклічного зсуву на 11 розрядів ліворуч над вхідними даними Q дорівнює Хемінговій вазі Q, отримуємо, що Хемінгова вага результату виконання операції $N'_1 = N_2 \oplus R(Q)$ буде залежати лише від значень N_2 і Q. Отже, з одного боку, елемент q_t не є відомим, а з іншого – можна довільно маніпулювати значеннями N_2, а також згідно з прийнятою моделлю витоку інформації можна отримати відомості про значення Хемінгової ваги блоку $N'_2 \| N'_1$.

Таким чином, при фіксованому t задача визначення значення вузла заміни k_t^h зведена до задачі встановлення входу K_t у значення h і виконання атаки на результат логічного додавання за модулем 2 значення невідомого вузла k_t^h з відповідною частиною підблоку N_2. Сформульована задача розв'язується шляхом проведення відомої інженерно-криптографічної атаки на реалізацію операції логічного додавання за модулем 2, описаної, наприклад, у [7, 8].

Визначення решти вузлів k_t^h проводиться шляхом зміни t і повторного виконання описаних вище дій. Після визначення значення усіх вузлів таблиці підстановки K проводиться визначення решти циклових ключів X_1, X_2, X_3, X_4, X_5, X_6, X_7.

3.3. Визначення циклових ключів X_1, X_2, X_3, X_4, X_5, X_6, X_7

Нагадаємо, що для успішного розв'язання задачі визначення циклового ключа необхідний контроль над аргументом N_1 операції логічного додавання за модулем 2^{32}. Після визначення циклового ключа X_0 і вузлів таблиці підстановки K можна побудувати рівняння оберненої функції першого циклу алгоритму криптографічного перетворення, які за відомими результатами виконання першого циклу N'_1 і N'_2 дають змогу знайти початкові значення N_1 і N_2.

Тому методика визначення циклового ключа X_1 є подібною до методики знаходження циклового ключа X_0 (див. п. 3.1) з тією відмінністю, що для встановлення необхідних значень бітів N'_1 необхідно враховувати рівняння оберненої функції першого циклу.

Аналогічно, для визначення циклових ключів X_2, X_3, X_4, X_5, X_6, X_7 необхідно будувати рівняння відповідних обернених циклів і враховувати їх для встановлення бітів N'_1 на початку кожного циклу.

Отже, використовуючи статистичні моделі операцій логічного додавання за модулем 2, за модулем 2^{32}, на основі інформації про споживану потужність комп'ютерних пристроїв, які реалізують алгоритм криптографічного перетворення на

підставі ГОСТ 28147-89, можна визначити таємні елементи цього криптографічного перетворення: ключ шифрування і вузли таблиці заміни.

4. Оцінка складності проведення інженерно-криптографічної атаки на реалізації алгоритму криптографічного перетворення за стандартом ГОСТ 28147-89

Оцінимо часову складність [11] проведення інженерно-криптографічної атаки на реалізацію алгоритму криптографічного перетворення за стандартом ГОСТ 28147-89, де за одиницю вимірювання приймемо кількість запусків пристрою для виконання шифрувань. Часовою складністю побудови рівнянь зворотних циклів знехтуємо, оскільки цю задачу можна розв'язати до проведення атаки, використавши підстановку визначених на попередніх етапах відомостей в узагальнену модель алгоритму криптографічного перетворення. Тоді часова складність визначення циклових ключів X_1, X_2, X_3, X_4, X_5, X_6, X_7 буде еквівалентна часовій складності C_0 визначення циклового ключа X_0. Загальна часова складність визначення циклових ключів становитиме

$$C_W = 8C_0. \tag{2}$$

Беручи до уваги, що кількість вузлів підстановки у кожній таблиці K_t є однаковою, то часова складність C_K визначення значень вузлів цілої таблиці K буде пропорційна до добутку значення часової складності C_t знаходження одного вузла таблиці на кількість вузлів у таблиці і кількість таблиць підстановки, тобто

$$C_K = 8*16C_t. \tag{3}$$

З врахуванням (2) і (3) часову складність визначення ключа шифрування і вузлів таблиці підстановки можна подати у вигляді

$$C = 8(C_0 + 16C_t). \tag{4}$$

Якщо для правильного визначення значення одного біту 32-бітового циклового ключа X_0 необхідно провести g тестових шифрувань, а для знаходження одного біту 4-бітового вузла підстановки слід здійснити m тестових шифрувань, то вираз (3) можна переписати у вигляді

$$C = 8(32g + 16*4m) = 256(g + 2m). \tag{5}$$

Врахувавши, що величини g і m є одного порядку, часову складність C із виразу (5) можна наближено записати у вигляді

$$C = 256(g + 2m) \approx 768g, \tag{6}$$

тобто часова складність визначення ключа шифрування і невідомих вузлів таблиці підстановки не залежить від значень цих елементів і є прямо пропорційна до кількості таємних бітів, використаних для криптографічного перетворення.

Згідно з результатами, наведеними у [4], для визначення одного біту невідомого аргументу необхідно виконати від 1000 до 5000 тестових шифрувань. Тоді оцінка (6) матиме значення в межах від одного до чотирьох мільйонів тестових шифрувань, що відповідає перетворенню від восьми до 32МБ відкритого тексту, враховуючи, що на одне тестове шифрування вимагається обробки восьми байтів вхідних даних. Така оцінка є значно кращою від оцінки складності атаки на основі аналізу розпису ключа [17, 18].

Висновки

Поряд із класичним математичним криптоаналізом, інженерно-криптографічні атаки на комп'ютерні реалізації криптографічних алгоритмів є ефективним засобом визначення таємної інформації, яка використана для перетворення даних. У даній статті розглянуто проведення такої атаки на комп'ютерну реалізацію алгоритму криптографічного перетворення згідно з чинним в Росії нормативним документом ГОСТ 28147-89, який прийнято за стандарт і в Україні. Запропонована атака ґрунтується на використанні інформації, яку можна отримати з одного із можливих

побічних каналів витоку інформації – споживаної потужності комп'ютерного пристрою.

Запропоновано методику проведення атаки на комп'ютерні реалізації алгоритму криптографічного перетворення за стандартом ГОСТ 28147-89, яка містить таку послідовність етапів: визначення першого циклового ключа, визначення невідомих вузлів таблиці підстановки і знаходження решти невідомих циклових ключів. Визначення таємних елементів цього криптографічного перетворення здійснюється з використанням статистичних моделей його складових операцій. Статистична модель операції логічного додавання за модулем 2^N для $N = 32$ використовується для обчислення циклових ключів, а статистична модель операції логічного додавання за модулем 2 – для знаходження вузлів таблиці підстановки. Це дозволяє повністю визначити таємні елементи, які використано при перетворенні даних – ключ шифрування і вузли таблиці підстановки.

З метою оцінки ефективності і перспектив практичної реалізації запропонованої атаки проведено оцінку часової складності проведення цієї атаки на комп'ютерні реалізації алгоритму криптографічного перетворення згідно зі стандартом ГОСТ 28147-89. Для здійснення запропонованої атаки вимагається значно меншого часу визначення ключа шифрування, ніж прямий перебір з відомими вузлами таблиці підстановки чи аналізу, що ґрунтується на особливостях процедури розпису ключа шифрування. Показано, що часова складність визначення ключа шифрування і вузлів таблиці підстановки не залежить від значень цих елементів і є прямо пропорційна до добутку кількості таємних бітів, використаних для криптографічного перетворення, на кількість тестових шифрувань, необхідних для знаходження одного невідомого біту.

Подальше зменшення часової складності можливе шляхом переривання виконання операцій шифрування, оскільки для знаходження таємних параметрів перетворення необхідно отримати дані про потужність, яка споживається при виконанні лише перших восьми циклів і одного додаткового повного шифрування для перевірки визначених ключа та вузлів таблиці шифрування.

Алгоритм криптографічного перетворення на підставі стандарту ГОСТ 28147-89 у режимі простої заміни є базовим перетворенням для інших режимів шифрування. Тому запропонована методика є фундаментом для створення засад побудови нових методик атакування комп'ютерних реалізацій інших режимів шифрування з використанням алгоритму криптографічного перетворення згідно зі стандартом ГОСТ 28147-89.

Отримані результати підкреслюють необхідність створення, розвитку і використання спеціальних методів, засобів проектування і продукування комп'ютерних реалізацій криптографічних алгоритмів, стійких до проведення інженерно-криптографічних атак з використанням даних, отриманих з побічних каналів витоку інформації. Запропоновану методику проведення інженерно-криптографічної атаки на комп'ютерні реалізації алгоритму криптографічного перетворення за стандартом ГОСТ 28147-89 можна використати для проектування, дослідження, тестування та сертифікації комп'ютерних засобів, які використовуються для оброблення конфіденційної інформації [19].

The methodic of attack execution on the computer realization of cryptographic transformation algorithm by the real standard in Ukraine was proposed. The estimation of timing complexity by the execution of this attack on the computer realization of cryptographic transformation algorithm was development. For the computing of ciphering key is used much less time than straight selection with known substitution table or analysis which based on the features of the ciphering key generating was proven.

Література

1. Kelsey J., Schneier B., Wagner D., Hall C. Side Channel Cryptanalysis of Product Ciphers // 5th European Symposium on Research in Computer Security – ESORICS '98, vol. 1485 of Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, 1998. – Pp. 97–110.
2. Clavier C., Coron J.-S., Dabbous N. Differential power analysis in the presence of hardware countermeasures // C.K. Koc, C.Paar, Eds., Cryptographic Hardware and Embedded Systems – CHESS 2000, vol. 1956 of Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, 2000. – Pp. 252–263.
3. Kocher P., Jaffe J., Jun B. Differential Power Analysis // Proceedings of International Conference CRYPTO'99, vol. 1666 of Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, 1999. – Pp. 388–397.
4. Messerges T., Dabbish E., Sloan R. Examining smart-card security under the threat of power analysis attack // IEEE Transactions on computers. – 2002. – Vol. 51. – No 5. – P. 541–552.
5. Messerges T., Dabbish E., Sloan R. Power analysis attacks of modular exponentiation in smartcards // C.K. Koc, C.Paar, Eds., Cryptographic Hardware and Embedded Systems – CHESS 1999, vol. 1717 of Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, 1999. – P. 144–157.
6. Akkar M., Giraud C. An implementation of DES and AES, secure against some attacks // In Proc. Cryptographic Hardware and Embedded Systems – CHES 2001, volume 2162 of Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, 2001. – P. 309-318.
7. Akkar M.-L., Bevan R., Dischamp P., Moyart D. Power analysis, what is now possible // T. Okamoto, Eds., International conference ASIACRYPT 2000, vol. 1976 of Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, 2000. – P. 489 – 502.
8. Messerges T. Using second-order power analysis to attack DPA resistant software // C.K. Koc, C.Paar, Eds., Cryptographic Hardware and Embedded Systems – CHESS 2000, vol. 1956 of Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, 2000. – P. 238 – 251.
9. ГОСТ 28147-89. Система обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. М.: Госстандарт СССР.
10. Коркішко Л.М., Васильцов І.В. Статистична модель операції додавання за модулем 2^N для проведення інженерно-криптографічних атак за побічними каналами витоку інформації // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2004. – № 8. – С. 115–121.
11. Черкаський М. Складність апаратно-комп'ютерних засобів // Матеріали міжнародної наук.-техн. Конф. "Сучасні проблеми в комп'ютерних науках в Україні" (CCU'2000). – Славське, 2000. – С. 58–67.
12. Feistel H. Cryptography and computer privacy // Scientific American. – 1973. – Vol. 228. – №5. – P. 15–23.
13. FIPS 46, "Data Encryption Standard", Federal Information Processing Standard (FIPS), Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington D.C.
14. National Institute of Standards and Technology (NIST), "Advanced Encryption Standard (AES) FIPS Publication 197," http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, Nov. 2001.
15. Коркішко Т.А., Мельник А. О., Мельник В.А. Захист інформації в комп'ютерних і телекомунікаційних мережах: Алгоритми та процесори симетричного блокового шифрування. – Львів: БАК, 2003. – 168 с.
16. Коркішко Т.А. Структурна організація алгоритмів симетричного блокового шифрування // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2001. – № 2. – С. 158–170.
17. Kelsey J., Schneier B., Wagner D. Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER and Triple-DES // N. Koblitz, Eds., International conference Advances in Cryptology – CRYPTO'96, vol. 1109 of Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, 1996. – P. 237 – 251.
18. Charnes C., O'Connor L., Pieprzyk J., Safavi-Naini R., Zheng Y. Comments on Soviet encryption algorithm // A. De Santis, Eds., International conference Advances in Cryptology – EuroCrypt '94, vol. 950 of Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, 1995. – P. 433 – 438.
19. Federal information processing standards publication. Security requirements for cryptographic modules. FIPS 140 – 2. National Institute of Standards and Technology (NIST), Gaithersburg, MD, 2001. – 68 p.

Одержано 07.04.2005 р.

.