УДК 004.056

СПОСІБ ПРОТИДІЇ РЕКОНСТРУКЦІЇ КЛЮЧІВ АЛГОРИТМУ ГОСТ 28147-89 АНАЛІЗОМ ДИНАМІКИ СПОЖИВАННЯ ПОТУЖНОСТІ

О. П. Марковський[1, а], В. Р. Чечель[2, б]

[1] *Національний технічний університет України "Київський політехнічний інститут"*
[2] *Національний університет кораблебудування імені адмірала Макарова*

Анотація

В роботі запропоновано та експериментально досліджено спосіб поліморфної реалізації алгоритму шифрування ГОСТ 28147-89 на мікроконтролерах і смарт-картах. Спосіб забезпечує захист від можливості реконструкції коду ключа алгоритму шляхом вимірювання та аналізу динаміки споживання потужності в процесі виконання алгоритму. Проведена експериментальна оцінка запропонованого способу.

Ключові слова: поліморфна реалізація, програмний поліморфізм, аналіз динаміки споживання потужності, смарт-карти, криптографічні алгоритми, протоколи захисту даних, термінальні обчислювальні пристрої

Вступ

Однією з домінуючих тенденцій сучасного етапу розвитку технологій комп'ютерної обробки даних є процес розширення та поглиблення інформаційної інтеграції. Забезпечуючи якісно більш високий рівень вирішення прикладних задач, інформаційна інтеграція охоплює всі рівні комп'ютерної обробки даних, включаючи вбудовані мікропроцесорні засоби, що вже зараз складають переважну частину термінальних пристроїв комп'ютерних мереж.

Важливою умовою ефективної взаємодії обчислювальних термінальних пристроїв в комп'ютерних мережах є надійна реалізація функцій захисту інформації, передбачених відповідними протоколами мережевого обміну даних. Основним засобом реалізації функцій захисту інформації в процесі мережевого обміну даними є криптографічні алгоритми. При реалізації цих алгоритмів на вбудованих мікроконтролерах і смарт-картах існує потенційна небезпека доступу до секретних елементів алгоритмів за допомогою вимірювання та аналізу параметрів технічної реалізації алгоритму на обчислювальному пристрої [1]. В процесі виконання програми на портативних обчислювальних пристроях існує зв'язок між командами, що виконуються, а також даними, з якими оперують команди, і значеннями деяких технічних параметрів пристрою. Найбільш інформативними в цьому плані є такі параметри, як споживана потужність та час виконання окремих фрагментів програми [2].

Це створює серйозну небезпеку для надійної реалізації захисту інформації при роботі портативних обчислювальних пристроїв в комп'ютерних мережах и вимагає розробки ефективних засобів протидії.

[а] markovskyy@i.ua
[б] viktoriyachechel@gmail.com

1. Аналіз відомих рішень і постановка задачі

До теперішнього часу створені розвинені технології реконструкції значень кодів окремих операндів за результатами вимірювання та аналізу динаміки споживання потужності мікроконтролерами в процесі реалізації ними алгоритмів захисту інформації [2].

На сьогоднішній день існує дві технології такої реконструкції: простий аналіз споживання потужності (SPA -Simple Power Analysis) та диференційний аналіз споживання потужності (DPA - Differentional Power Analysis) [3].

Сутність першої полягає у відновленні за осцилограмою споживання потужності послідовності команд, що виконуються програмою. Ефективність SPA в плані реконструкції даних визначається залежністю порядку виконання програми від цих даних.

Технологія DPA полягає у встановлені статичних залежностей між розрядами даних та потужністю, що споживається обчислювальним пристроєм під час виконання кожної з команд. Тобто ця технологія попереднього статистичного дослідження впливу розрядів даних на споживання потужності в кожний момент часу за умови, що програма не змінюється. Відповідно, такий процес може бути ефективним лише за умови незмінності даних.

Активне використання технологій SPA і DPA ініціює розробку апаратних та програмних засобів протидії. Застосування апаратних засобів ускладнює структуру портативних обчислювальних компонент та помітно збільшує їх вартість. Тому на практиці більшого розповсюдження набули програмні засоби протидії SPA та DPA. Основними критеріями ефективності таких засобів є рівень захисту, що забезпечується їх застосуванням та об'єм додаткових обчислювальних ресурсів, потрібних для їх реалізації.

Для протидії DPA - технології, що має за основу статистичний аналіз, найбільш ефективними є методи, що базуються на введенні випадковості.

До цієї групи методів протидії відносяться:

- маскування значень даних, що використовуються при кожному виконанні програми (ключів криптографічних алгоритмів) випадковими кодами, які змінюються при кожному запуску програми. В останні роки з'явилась технологія незаконного доступу до даних - диференційний аналіз споживання потужності (DPA) другого порядку, при використанні якої маскування ключів випадковим кодом не забезпечує їх ефективного захисту від реконструкції.
- стохастичний програмний поліморфізм, що застосовується в двох формах: випадкова вставка команд, що не впливають на результат, але ускладнюють прив'язку моменту виміру потужності до конкретної команди програми; випадкова зміна послідовності виконання незалежних по даним команд.

В більшості країн СНД в якості стандарту симетричного шифрування використовується алгоритм ГОСТ 28147-89 [4, 5]. Цей алгоритм являє собою шифроблок, що виконує обробку 64-розрядного блоку даних $D = \{d_1, d_2, ..., d_{64}\}, \forall\ h \in \{1, ..., 64\} : d_h \in \{0, 1\}$ з використанням 256-розрядного ключа K. На виході формується 64-розрядний код $C = \{c_1, c_2, ..., c_{64}\}$.

Відомі реалізації цього алгоритму [6] з використанням маскування для захисту від DPA. При маскуванні проміжних результатів програмної реалізації алгоритму ГОСТ 28147-89 може використовуватися різна кількість масок. Найбільш ефективним є варіант, за яким, додатково до маски Y даних, використовуються маски X для маскування ключів. При цьому кількість різних масок X може співпасти з кількістю циклів (32) або з кількістю субключів (8). Стосовно алгоритму ГОСТ 28147-89 проблема зняття маски ускладнюється тим, що в цьому алгоритмі використовуються як арифметичні, так і логічні операції.

Недоліком цих реалізацій є те, що вони потребують значних ресурсів пам'яті для зберігання спеціальних таблиць для зняття маски. Зокрема, в опублікованих варіантах [6] маскування операндів алгоритму ГОСТ 28147-89 в процесі його виконання об'єм пам'яті таблиць становить до 2^{31} бітів. При цьому час реалізації алгоритму збільшується приблизно вдвоє. При менших об'ємах пам'яті втрата продуктивності алгоритму ще більш значна.

Крім того, маскування не забезпечують захисту ключа від його реконструкції застосуванням технологій DPA другого порядку.

Ціль досліджень полягає в розробці ефективного способу поліморфної реалізації алгоритму ГОСТ 28147-89 захищеної від DPA.

2. Спосіб поліморфної реалізації ГОСТ 28147-89

Найбільш ефективним засобом протидії SPA і DPA є друга форма програмного поліморфізму, тобто випадкова зміна при кожному виконанні програми послідовності виконання команд, яка не впливає на результат. Такий поліморфізм дозволяє ефективно протидіяти DPA високих порядків [3].

Виходячи з того, що диференційний аналіз базується на встановленні кореляційних залежностей між потужністю, споживаною обчислювальним пристроєм в кожний момент виконання програми та розрядами ключа, найбільш дієвим способом протидії такому аналізу є стохастична зміна порядку слідування команд при кожному виконанні програми, тобто поліморфна реалізація алгоритмів симетричного шифрування.

Застосування симетричних алгоритмів передбачає незалежне шифрування кожного з блоків, на які розділяється повідомлення. Це дозволяє організувати квазі паралельну обробку декількох блоків і, тим самим, створює передумови для реалізації поліморфізму на рівні обробки блоків даних.

Таким чином, верхній рівень поліморфної реалізації алгоритмів симетричного шифрування полягає в стохастичному переключенні з обробки одного блоку даних на обробку іншого.

Нижній рівень поліморфної реалізації алгоритмів симетричного шифрування даних полягає в тому, що змінюється порядок виконання команд обробки одного блоку.

Аналіз обчислювальних процедур алгоритму ГОСТ 28147-89 показує, що найбільш вразливим для DPA є нелінійні перетворення, що послідовно виконуються над 4-розрядними фрагментами арифметичної суми даних та поточного субключа. Невелика розрядність операндів дозволяє достатньо просто перебрати всі 16 варіантів 4-розрядного коду, що приймає участь в цій операції. Оскільки аналіз фрагментів ключа можна виконувати незалежно, то всього потрібно проаналізувати $16 \cdot 64 = 1024$ варіанти.

При реалізації алгоритму ГОСТ 28147-89 на процесорах, розрядність яких менша 32, то з позицій вразливості до SPA, існує проблема безпечної реалізації циклічного зсуву на 11 розрядів без використання команд умовних переходів та змінних, що приймають два значення - 0 та 1.

В рамках проведених досліджень було реалізовано поліморфне виконання операцій табличного перетворення для архітектури мікроконтролеру Intel 8051. В цих операціях дані, що містять частину ключової інформації виставляються на шину мікроконтролера для звернення до пам'яті, в якій зберігаються таблиці нелінійних перетворень. Інші операції - регістрові і споживання потужності при їх виконанні суттєво менше, ніж при звертанні до пам'яті. В силу того, що операції табличних перетворень незалежні між собою за даними, порядок їх виконання може бути довільним. Фактично в рамках проведених до-

сліджень було реалізовано 24 варіанти табличного перетворення на кожному з 32-х циклів виконання алгоритму ГОСТ 28147-89.

Для реалізації поліморфізму при виконанні табличних перетворень замість випадкового виконання послідовності звернень до таблиць, виконано циклічне переключення послідовності з випадковим вибором початкової операції. Це зумовлено обмеженнями на об'єм регістрової пам'яті мікроконтролера. Для управління поліморфним виконанням нелінійних табличних перетворень використовується десять команд .

Для реалізації генератора псевдовипадкових чисел, що використовуються для вибору послідовності обробки фрагментів даних при їх нелінійному перетворені використано лінійний зсувний регістр з лінійною функцією зворотного зв'язку (LFSR-Linear Feedback Shift Register). при цьому стартовий код та код лінійної функції зворотного зв'язку вибирається з певного набору в процесі ініціалізації. Для організації роботи такого генератора в програмі використано всього 6 команд.

Проведені експериментальні дослідження показали, що організація поліморфного виконання нелінійних перетворень при реалізації алгоритму ГОСТ 28147-89 має наслідком зменшення продуктивності в 1.7 раз.

Для оцінки ефективності протидії DPA розробленого варіанту поліморфної реалізації алгоритму ГОСТ 28147-89 був розроблений комплекс симуляції для архітектури Intel 8051, яка широко використовується в сучасних смарт-картах. Оцінка проводилася по класичній моделі Хемінгової відстані, згідно з якою споживана потужність визначається кількістю змінюваних бітів коду (схеми на основі КМОП-технології).

На вибіркових значення (10^4) ключів експериментально доведено, що максимальне значення коефіцієнту кореляції споживаної потужності та значень бітів ключа при запропонованій поліморфній реалізації алгоритму ГОСТ 28147-89 становило 0,0452 на противагу значенню 0,1648 при традиційній реалізації алгоритму. Таким чином, запропонована поліморфна реалізація алгоритму ГОСТ 28147-89 дозволяє в чотири рази зменшити залежність споживаної потужності мікроконтролера від значень бітів ключа.

Висновки

В результаті проведених досліджень розроблено та експериментально досліджено спосіб поліморфної реалізації алгоритму ГОСТ 28147-89 на мікроконтролерах архітектури Intel 8051. Спосіб полягає в організації виконання найбільш вразливих для DPA операцій нелінійних перетворень. Експериментально доведено, що розроблена реалізація дозволяє в 4 рази зменшити залежність споживаної потужності від бітів ключа за рахунок зменшення продуктивності шифрування в 1.7 рази.

Перелік використаних джерел

1. Akkar M. L., Bevan C., Dischamp P., Moyart D. Power analysis, what is now possible // Proceeding of International Workshop "Asiacrypt-2000". LNCS-1976. Springer-Verlag. — 2000. — p. 489-502.
2. Messerges T. S., Dabblish E. A., Sloan R. H. Examining Smart-Cart Sequruty under the threat of Power-Analysis Attacks // IEEE Transaction on Computers. — Vol. 51. — No. 5. — 2002. — p. 541-522.
3. Biham E., Shamir A. Power Analysis of the Key Scheduling of the AES Candidates // Proc. of 2-th AES Candidate Conference.Roma. — 1999. — p. 115-121.
4. Винокуров А. Ю. Гост не прост . . . , а очень прост! // Монитор — 1995. — №1. — с. 60-73.
5. ГОСТ 28147-89. Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. М. 1989.
6. Абабне О. А., Марковский А. П., Ияд Мохд Маджид Ахмад Шахрури Способ защиты ключей алгоритма ГОСТ 28.147-89 от реконструкции анализом динамики потребляемой мощности // Вісник національного технічного університету України "КПІ". Інформатика, управління та обчислювальна техніка. — 2007. — №46. — с. 128-138.