© *Karpinski M.,Korkishko L. DAP attack modelling of first degree // Ukrainian Scientific Journal of Information Security, 2016, vol. 22, issue 2, p. 184-195.*

МОДЕЛЮВАННЯ DPA АТАКИ ПЕРШОГО ПОРЯДКУ

Микола Карпінський[1], Леся Коркішко[2]

[1]Університет у Бєльсько-Бялій, Польща
[2]Тернопільський національний технічний університет ім. І. Пулюя, Україна

КАРПІНСЬКИЙ Микола Петрович, д.т.н.

Рік та місце народження: 1958 рік, м. Балей, РФ.
Освіта: Львівський політехнічний інститут (з 2000 року – Національний університет «Львівська політехніка»), 1980 рік.
Посада: завідувач кафедри інформатики та автоматики з 2015 року.
Наукові інтереси: безпека інформаційних технологій.
Публікації: більше 100 наукових публікацій, серед яких монографії, навчальні посібники, наукові статті, матеріали конференцій та патенти на винаходи.
E-mail: mkarpinski@ath.bielsko.pl

КОРКІШКО Леся Мирославівна

Рік та місце народження: 1974 рік, м. Червоноград, Україна.
Освіта: Національний університет «Львівська політехніка», 1997 рік.
Посада: науковий співробітник НДЧ кафедри кібербезпеки з 2016 року.
Наукові інтереси: інформаційна безпека
Публікації: 20 наукових публікацій, серед яких розділи у монографіях, наукові статті, матеріали та тези доповідей на конференціях.
E-mail: lesykkor@yahoo.com

Анотація. *У даній роботі проведено моделювання DPA-атаки першого порядку на основі кореляційних коефіцієнтів на HDL-моделі криптографічних процесорів за алгоритмами ГОСТ 28147-89 та mCrypton, які обробляють дані у масковану представленні. Для цього створено систему моделювання DPA-атаки, яка включає у себе аналізатор споживаної потужності, на який подають попередньо отримані дані про паразитні взаємовпливи елементів та з'єднань, часово-анотовану схему процесора, файл VCD (внутрішньої активності елементів схеми). У результаті отримують розгорнуті в часі траси споживаної потужності, які є вхідними даними для DPA-атаки першого порядку на основі кореляційних коефіцієнтів. Виходячи з результатів моделювання цієї атаки на HDL-моделі ядер криптографічних процесорів та особливості архітектури розглянутих процесорів, показано, що ці процесори будуть володіти підвищеною стійкістю до цієї атаки. Характеристики розроблених ядер процесорів дозволяють рекомендувати їх до використання у пристроях з обмеженими ресурсами (смарт-карти, криптографічні токени, мобільні пристрої зв'язку, тощо), які будуть мати підвищену стійкість до DPA-атак першого порядку.*

Ключові слова: *захист інформації, DPA-атаки, ядра процесорів шифрування, масковане представлення даних, канал витоку інформації.*

Вступ

Алгоритми криптографічних перетворень широко застосовуються у сучасних комп'ютерних системах для вирішення задач захисту інформації. Серед таких алгоритмів виділяють симетричні алгоритми шифрування, які використовуються для шифрування даних користувача з подальшим їх передаванням незахищеними каналами зв'язку чи записуванням їх на незахищені пристрої збереження інформації. Основними способами реалізації цих алгоритмів є програмний спосіб на основі програмованих універсальних процесорів та апаратний – на основі спеціалізованих процесорів. Основними перевагами програмної реалізації криптографічних алгоритмів є їх відносна простота кодування, налагодження та оновлення, низька вартість реалізації. Однак апаратна реалізація алгоритмів широко використовується у застосуваннях, де необхідно задовольняти жорсткі вимоги в частині споживаної потужності пристрою та продуктивності обробки даних. При цьому складність розробки, налагодження та виготовлення апаратних реалізацій алгоритмів є значно вищою за програмні реалізації.

Реалізація алгоритмів шифрування на комп'ютерних платформах часто створює передумови для витоку інформації про конфіденційні дані, які обробляються, зокрема про ключі шифрування. Такий витік інформації зумовлюється фактом існування побічних каналів її витоку із комп'ютерних реалізацій криптографічних пристроїв (часу виконання алгоритмів, електромагнітного випроміню-

вання, споживаної потужності, тощо) та існування відповідних класів інженерно-криптографічних атак, які використовують ці побічні канали [1].

Враховуючи існування побічних каналів витоку інформації, реалізація криптографічних процесорів, які володіють підвищеною стійкістю до таких атак, може потребувати кількох ітерацій, які включають у себе такі етапи: оцінки стійкості зразка процесора до заданого переліку атак, внесення змін до архітектури процесора, модифікування опису процесора, та виготовлення нового зразка. Так як одна така ітерація потребує створення нової проектної документації, виготовлення нових технологічних компонент та перезапуску напівпровідникового виробництва для створення нової малосерійної партії пристроїв, її вартість є достатньо високою а тривалість може складати до двох місяців. Тому актуальною є задача оцінки стійкості апаратно-орієнтованого процесора реалізації криптографічних алгоритмів ще до моменту передачі технологічної документації у виробництво.

У даній роботі запропоновано проводити оцінку захищеності процесорів симетричного блокового шифрування від атак на основі аналізу споживаної потужності на етапі їх проектування. Така оцінка захищеності не буде однозначно давати відповідь на питання про захищеність реального виробу, що зумовлено потенційними не відповідностями моделей напівпровідникових компонентів та їх міжз'єднань з фактичними характеристиками цих компонентів, на які додатково впливають варіації параметрів технологічного процесу. Разом з тим, вона дозволить виявити помилки та неточності ще на етапі проектування, що дозволить значно скоротити час та знизити вартість розробки процесорів. Для такого аналізу авторами обрано два процесори, які реалізують алгоритми симетричного блокового шифрування за алгоритмами ГОСТ 28147-89 [2] та mCrypton [3] та обробляють дані у маскованому представленні, що дозволяє створювати системи захисту інформації з підвищеною стійкістю до атак на основі аналізу споживаної потужності.

DPA атака на основі кореляційних коефіцієнтів

У відкритих джерелах опубліковано декілька типів атак на пристрої на основі диференційного аналізу споживаної потужності – так званих DPA атак. Серед них: DPA атака на основі різниці середніх [4], DPA атака на основі відстані середніх [5], узагальнене тестування на найбільшу вірогідність [6], DPA атака на основі зразків [7]. Разом з тим, базовим методом атак залишається DPA атака на основі аналізу кореляційних коефіцієнтів [8]. Тому, у даній роботі було обрано саме цю атаку для оцінки захищеності процесорів симетричного блокового шифрування на етапі їх проектування.

Стратегію DPA-атаки на основі аналізу кореляційних коефіцієнтів можна подати у вигляді кількох етапів [5]. На першому етапі обирається деяке проміжне значення $f(d,k)$ у графі обчислень криптографічного алгоритму, яке буде обчислене даним пристроєм, де d – відомі дані (відкритий текст чи шифр текст), які можна довільно змінити, маніпулюючи входом пристрою, k - елемент ключа.

На другому етапі здійснюють вимірювання споживаної потужності пристрою у процесі зашифрування чи розшифрування D різних блоків даних. Для кожного з цих зашифрувань чи розшифрувань аналітику необхідно знати відповідні значення d, які при цьому утворюються та беруть участь у обчисленні $f(d,k)$. Позначимо такі відомі значення, як $\bar{d} = (d_1,\ldots,d_D)^T$, де d_i - відомі дані у i -му зашифруванні чи розшифруванні.

Протягом кожного з таких зашифрувань чи розшифрувань аналітик записує сигнал споживаної потужності – так звану «трасу». Кожна траса відповідає певному блоку d_i та позначається як $\bar{t}_1 = (t_{i,1},\ldots,t_{i,T})$, де T - довжина траси у відліках. Тому набір трас можна подати у вигляді матриці $|W|$ розміру $D \times T$. Зауважимо, що для DPA атаки є суттєвим правильне вирівнювання трас у часі: виміряні значення споживаної потужності у кожній колонці t_j матриці $|W|$ повинні спричинятися однією і тією ж операцією.

На третьому етапі обчислюють очікувані проміжні значення для кожного можливого елементу ключа k_j вектора $\bar{k} = (k_1,\ldots,k_K)$, де K – загальна кількість комбінацій варіантів вибору різних k_j. Такі k_j називають гіпотезами ключа. Тоді аналітик обчислює матрицю $|V|$ усіх можливих варіантів проміжних значень $v_{i,j} = f(d_i,k_j)$, де $i = 1,\ldots,D$, $j = 1,\ldots,K$. Так як $\bar{k}$ містить усі можливі варіанти елемента ключа, то метою аналітика є визначення індексу стовпця $|V|$, який відповідає елементу ключа, який був використаний при обробці $\bar{d}$.

На четвертому етапі проводиться відображення $|V|$ у матрицю $|H|$ очікуваних значень споживаної потужності: $v_{i,j} \to h_{i,j}$. Для цього аналітик використовує відомості про модель споживаної потужності, яка характерна для досліджуваного пристрою. Найбільш розповсюдженими є моделі на основі ваги Хеммінга, відстані Хемінга та так звані нуль-моделі.

На останньому етапі кожен стовпець h_i порівнюють із кожним стовпцем t_j, тобто порівнюють очікувані значення споживаної потужності для кожної гіпотези елементу ключа із фактично отриманими трасами для деякого елементу ключа. Результатом такого порівняння є матриця $|R|$ кореляційних коефіцієнтів $r_{i,j}$, де $i = 1,\ldots,K$, $j = 1,\ldots,T$. Індекси найбільших коефіцієнтів матриці $|R|$ вказують на індекси елементу ключа, який був використаний пристроєм при обробці $\bar{d}$.

Методи захисту від DPA атак

Базовим методом захисту від проведення DPA атак є модифікування споживаної потужності крип-

тографічного пристрою таким чином, щоб вона не залежала від проміжних даних, які обробляються. Аналіз таких опублікованих методів захисту дозволяє виділити у них три основні групи.

Перша група методів полягає у використанні неоднакових ключів шифрування для різних сесій обробки даних обмеженого обсягу у криптографічному пристрої. Тоді аналітик володіє обмеженим обсягом даних, які він може використати для проведення атаки. Обсяг необхідних даних обирається таким чином, щоб його не вистачало для успішного атакування криптографічного пристрою. Недоліком наведеної групи методів є обмеженість її застосування, оскільки лише невелика кількість криптографічних протоколів підтримує часту зміну ключів шифрування.

Друга група методів зводиться до усунення чи зменшення залежності значення споживаної потужності від значень даних, які обробляються. Реалізація таких методів полягає у:

– рандомізуванні виконання алгоритму (часовий аспект) шляхом виконання базових операцій алгоритму у різні (рандомізовані) моменти часу;

– такій зміні споживаної потужності елементів при обробці різних даних (амплітудний аспект), при якій задача виявлення цих залежностей для аналітика значно ускладнюється.

Рандомізування виконання алгоритму криптографічного перетворення досягається двома способами: випадковою зміною шляху виконання алгоритму та перемішуванням порядку операцій. Для випадкової зміни шляху виконання алгоритму перед кожним його виконанням генерується випадкове число і на його основі обираються місця вставляння порожніх операцій та кількість порожніх операцій. Особливістю цього способу є зменшення продуктивності обробки даних внаслідок збільшення критичного шляху алгоритму криптографічного перетворення із-за додаткових порожніх операцій. Прикладами реалізації цього способу є [9, 10]: вставляння порожніх операцій між операціями обробки даних, вставляння порожніх циклів очікування, випадковий пропуск тактових імпульсів, випадкова зміна значення частоти тактових імпульсів, випадкове перемикання між декількома тактовими сигналами з різною частотою [11].

Перемішування операцій алгоритму криптографічного перетворення полягає у випадковій зміні порядку виконання складових операцій алгоритму [12, 13]. При цьому критичний шлях алгоритму залишається незмінним. Особливістю застосування цього способу є залежність кількості варіантів перемішування від структури чи рівня паралелізму алгоритму криптографічного перетворення. Самостійно метод перемішування операції не використовується, оскільки його вплив нейтралізується шляхом використання так званого вирівнювання трас споживаної потужності у часі [14]. Тому на практиці використовується комбінування розглянутих двох способів.

При амплітудному вирівнюванні зміна споживаної потужності досягається такими способами [15-17]: збільшенням рівня шуму у сигналі про споживану потужність та вирівнюванням споживаної потужності компонент при обробці різних даних. Рівень шуму у сигналі про споживану потужність збільшують за допомогою під'єднання генератора шуму до ліній живлення пристрою, використання додаткових паралельно працюючих обчислювальних блоків, які обробляють випадкові дані [18, 19].

Вирівнювання споживаної потужності компонент криптографічного пристрою досягається шляхом: а) використання спеціально підібраного набору інструкцій процесора, у яких є збалансована вага Хемінга для виконання операцій над різними даними; б) застосування спеціального стилю програмування – уникнення умовних переходів, які використовують елементи ключа шифрування як умову переходу, уникнення генерування адрес пам'яті, які залежать від елементів ключа шифрування, тощо; в) побудови криптографічного пристрою на базі спеціальних DPR (Dual-Rail Precharge) логічних вентилів із вирівняним споживанням потужності, наприклад, елементів SABL (Sense Amplifier Based Logic) [20], WDDL (Wave Dynamic Differential Logic) [21], DSDR (Dual-Spacer Dual-Rail) [22-24] та їх варіантів TDPL (Three-Phase Dual-Rail Precharge Logic) [25], 3sDL (3-state Dynamic Logic) [26]. Також до цього способу відносять фільтрування ліній живлення пристрою за допомогою активних і пасивних фільтрів [27-29] та використання асинхронної логіки, наприклад [30-34]. Аналіз методів усунення залежності споживаної потужності від даних, які обробляються у криптографічному пристрої, показав, що поодинці ці методи не дозволяють надійно захистити ці засоби від атак на основі аналізу споживаної потужності. Однак, зазначені методи доцільно використовувати сумісно з іншими методами захисту для збільшення складності атаки криптографічних пристроїв.

До третьої групи методів відноситься рандомізування проміжних значень, які обробляються криптографічним пристроєм. Цей метод можна застосовувати на кількох рівнях структури криптографічного пристрою: на рівні способу виготовлення пристрою, на рівні проміжних результатів які отримуються в процесі роботи пристрою. На рівні способу реалізації криптографічного пристрою застосовують рандомізовані логічні елементи та рандомізовані компоненти криптографічного пристрою. На рівні проміжних результатів процесу виконання алгоритму криптографічного перетворення використовують масковане представлення даних на основі технології розділення таємниці [35- 37].

Основою рандомізування є подання даних чи сигналів a у маскованому представленні у вигляді пари $\{\tilde{a}, x\}$, причому $\tilde{a} = a \,\mathrm{op}\, x$, op – бінарна операція, яка належить скінченому полю чи кільцю, x – маска, випадкове число з рівномірним розподілом ймовірності, незалежне від a. Дані a називають відкритими даними, $\tilde{a}$ – замаскованими даними, а пара $\{\tilde{a}, x\}$ – це масковані дані. Маска x генерується всередині криптографічного пристрою, наприклад, за допомогою генератора випадкових чисел, та є невідомою аналітику. Вибір операції маскування залежить від операцій, які використовуються в алгоритмі криптографічного перетворення. Типовими

операціями маскування є додавання за модулем два, додавання за модулем 2^N.

Беручи до уваги, що специфікації алгоритмів криптографічного перетворення створені для опису процесу обробки немаскованих даних, то для обробки даних у маскованому представленні ці специфікації повинні бути зміненими. Необхідні зміни у специфікацію алгоритму вносять для того, щоб врахувати факт подання оброблюваних даних у маскованому вигляді. З метою захисту від атак на основі аналізу споживаної потужності усі проміжні дані, які утворюються в процесі обробки даних, теж подають у маскованому представленні, а процес обробки даних повинен бути організованим таким чином, щоб уникнути використання чи появи немаскованих даних на будь-якому етапі обчислень. Паралельно з обробкою маскованих даних обробляють і саму маску, здійснюючи так званий процес корекції маски. Після виконання обчислень з використанням конфіденційної інформації, дані переводяться із маскованого представлення у немасковане шляхом виконання операції маскування над маскованими даними та оберненим елементом маски. У загальному випадку, маскування даних може відбуватися з використанням будь-якої кількості масок на основі техніки розділення таємниці.

Рандомізовані логічні елементи використовують для створення компонентів криптографічних пристроїв, які обробляють дані з використанням конфіденційної інформації. Робота таких елементів ґрунтується на застосуванні методу маскування до електричних сигналів, які обробляються. Прикладом таких логічних елементів, що отримали назву маскові логічні елементи, є MDPL (Masked Dual-Rail Precharge Logic) [38, 39]. Елементи MDPL використовують одну маску для обробки усіх маскованих вхідних даних. Альтернативні варіанти елементів MDPL були розроблені у [40]. Перевагою цих елементів є їх стійкість до гонок сигналів, що в свою чергу дозволяє уникнути залежності споживаної потужності від оброблюваних даних [41]. Однак недоліком використання елементів MDPL є необхідність розробки криптографічного пристрою на основі напівзамовлених інтегральних схем із спеціальними бібліотеками елементів, що призводить до значного збільшення вартості цих пристроїв. Альтернативні підходи до побудови маскованих логічних елементів на базі стандартних бібліотечних компонентів були запропоновані у [42] для апаратної і програмної реалізації та розвинуті у [43] для апаратної реалізації. У [44] запропонований варіант маскованих логічних елементів, однак використана для їх побудови базова модель не враховує можливість виникнення гонок.

Рандомізовані апаратні компоненти криптографічних пристроїв використовують розглянутий вище принцип маскування. При цьому виділяються закінчені апаратні блоки, наприклад, маскований помножувач [45], масковані шини обміну даними [46-48], рандомізовані елементи пам'яті на основі записування випадкових значень на місця немаскованих даних [49]. Особливістю застосування рандомізованих апаратних компонентів є збільшення апаратної, часової та місткістної складності цих компонентів у порівнянні з їх немаскованими аналогами.

Архітектура криптографічних процесорів

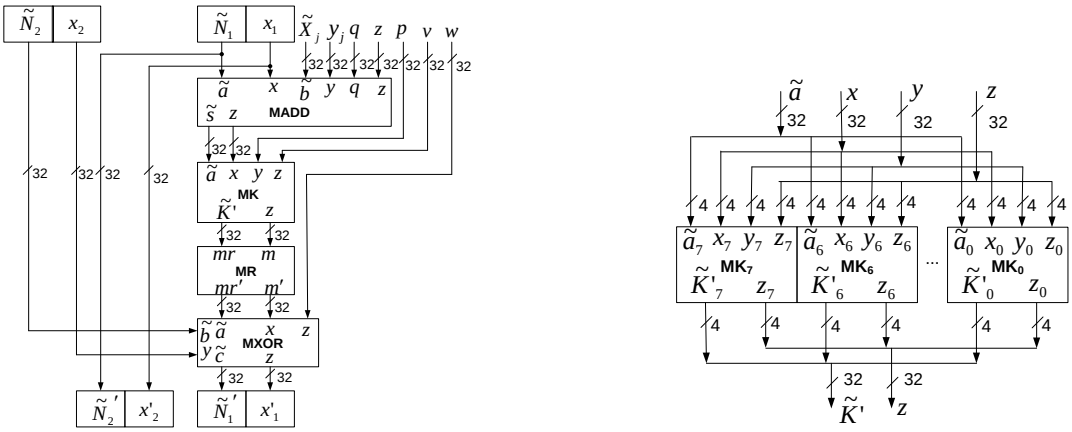
Архітектура криптографічних процесорів основана на апаратному відображенні одного циклу алгоритму шифрування із паралельним генеруванням циклового ключа та обробці даних у маскованому представленні [50]. Обидва процесори включають у себе такі функціональні блоки: блок інтерфейсу, блок пам'яті ключів, блока пам'яті таблиці заміни, блок обробки даних, блок оновлення таблиці замін, пристрій керування та генератор випадкових чисел. Обробка даних здійснюється блоком обробки даних з використанням маскованих циклових ключів та їх відповідних масок, маскованої таблиці заміни та її відповідної маски. Для обробки даних і оновлення таблиці заміни використовуються випадкові числа з виходу генератора випадкових чисел. Структура пристрою обробки даних у маскованому представленні та блока пам'яті таблиці заміни для алгоритму ГОСТ 28147-89 [51] наведено на рис. 1.

На вхід блоку обробки даних надходять дані у вигляді маскованих блоків $\tilde{N}'_2$ і $\tilde{N}'_1$ та відповідні маски x'_2 і x'_1. Для обробки даних використовуються масковані циклові ключі $\tilde{X}_j$ та відповідні маски y_j з блоку пам'яті ключів, 32-розрядні випадкові числа q, z, p, v, w від генератора випадкових чисел. Ліва частина маскованого блоку з відповідною маскою посилаються на вхід маскованого суматора за модулем 232 [52]. На другий вхід цього суматора подано маскований цикловий ключ з відповідною маскою. На третій і четвертий входи суматора подано випадкові числа q і z. Результат додавання частини блоку маскованих даних та маскованого ключа разом з відповідною маскою подаються на адресний вхід маскованої таблиці підстановки MK. На додаткові входи MK подаються випадкові числа p і v. Блок пам'яті таблиць замін організований як набір із восьми 4-розрядних таблиць замін $\tilde{K}'_i$, $0 \le i \le 7$ (рис. 1б). Результат виконання таблиці підстановки із відповідною маскою подаються на входи блоку циклічного зсуву MR. Результат циклічного зсуву маскованих даних та маски на 11 розрядів ліворуч поступає на перший вхід суматора за модулем два маскованих даних $MXOR$. На другий вхід цього суматора подаються ліва частина маскованого блоку даних $\tilde{N}_2$ і відповідна маска x_2.

На третій вхід суматора подано випадкове число w. Результат виконання операції додавання за модулем два та маска результату присвоюються правій частині результату циклу $\tilde{N}'_1$ та правій частині маски x'_1. Описаний процес повторюється в кожному із заданої кількості циклів.

Під час обробки даних блок оновлення таблиці замін оновлює вміст блоку пам'яті таблиці замін на основі проміжної маски p та вихідної масок v і початкової маски таблиці z'. Таблиця заміни онов-

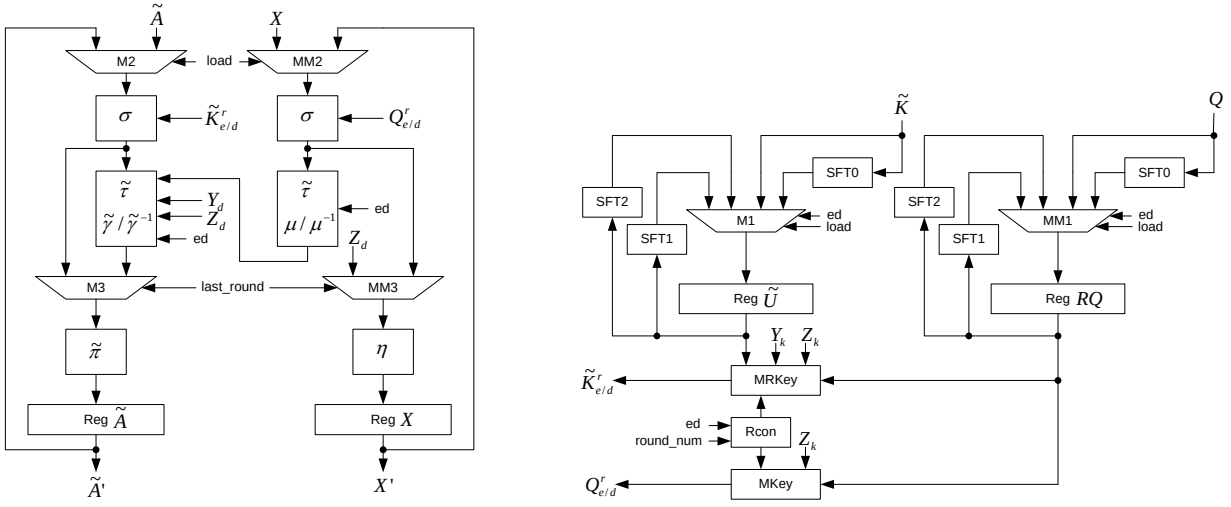
люється із заданою періодичністю в процесі обробки даних чи в процесі ініціалізації обчислень.



а) б)

Рис. 1. Структурні схеми процесора ГОСТ 28147-89: а) блоку обробки маскованих даних, б) блоку таблиці підстановки для даних у маскованому представленні

Аналогічно, блок обробки маскованих даних за алгоритмом mCrypton [53] реалізує один цикл шифрування та алгоритм обчислення чергового циклового ключа. Дані A та ключ шифрування K задаються у маскованому представленні – $\tilde{A} = A \oplus X$ і $\tilde{K} = K \oplus Q$ відповідно, де X і Q є незалежні випадкові числа. Отримані дані A' є поданими у маскованому представленні $\tilde{A}' = A' \oplus X'$, де X' є маскою результату. Блок обробки даних процесора використовує масковані циклові ключі $\tilde{K}^r_{e/d} = K^r_{e/d} \oplus Q^r_{e/d}$ та відповідні маски циклових ключів $Q^r_{e/d}$, згенеровані блоком обробки ключа (рис. 2).



а) б)

Рис. 2. Структури процесора mCrypton: а) блоку обробки маскованих даних, б) блоку обробки маскованого ключа

Для виконання маскованих операцій $\tilde{\gamma}$ і $\tilde{\gamma}^{-1}$ використовуються додаткові маски Y_d і Z_d. Ця ж маска Z_d використовується у частині обчислення корекції маски. Зауважимо, що у даній структурі реалізація операцій $\tilde{\gamma}$ / $\tilde{\gamma}^{-1}$ та $\tilde{\tau}$ є об'єднаними у один блок із набором 16 таблиць підстановок та відповідних мультиплексорів. Відповідні перетворення μ / μ^{-1} і $\tilde{\tau}$ виконано на основі мультиплексорів, які працюють синхронно із $\tilde{\gamma}$ / $\tilde{\gamma}^{-1}$ та $\tilde{\tau}$.

Блок обробки ключа генерує масковані циклові ключі та відповідні їм маски на основі заданого ключа шифрування та режиму шифрування. Аналогічно до блоку обробки даних, блок обробки ключа складається з двох частин – обробки маскованого ключа і генерування маскованих циклових ключів $\tilde{K}^r_{e/d}$ та частини обробки та генерування відповідних масок циклових ключів $Q^r_{e/d}$.

Спочатку маскований ключ та маска ключа записуються у відповідні регістри Reg $\tilde{U}$ і Reg RQ. Далі значення цих регістрів використовуються блоками MRKey і MKey (разом із цикловими константами $C[r]$ з блоку Rcon) для генерування маскованих циклових ключів та їх масок. Блок MRKey додатково використовує випадкові маски Y_k і Z_k для прове-

дення перетворень із маскованими S-таблицями. Ця ж маска Z_k використана у блоці MKey для обчислення відповідної маски циклового ключа. Регістри маскованого ключа та маски ключа оновлюються до своїх нових значень за допомогою циклічних зсувів SFT1 чи SFT2.

Реалізація маскованих таблиць підстановок вимагає їх регулярного оновлення перед кожним використанням із новими масками. З точки зору безпеки обчислень, алгоритм шифрування mCrypton є вразливим на етапах обчислень першого та останнього циклу. Тому у цьому процесорі пристрій керування здійснює обчислення маскованих таблиць підстановки лише для першого та останнього циклів. Інші цикли використовують вже обчислені масковані таблиці підстановки. Однак, для зменшення ймовірності встановлення деяких проміжних масок у нульове значення, пристрій керування обирає випадковим чином деякий додатковий номер циклу та ініціює для нього обчислення таблиць підстановок із новими масками Y_k, Z_k, Y_d, та Z_d. Такий прийом дозволяє знизити вимоги до продуктивності генератора випадкових чисел та збільшити продуктивність роботи процесора в цілому.

Система моделювання DPA атаки на HDL моделі криптографічних процесорів

Система моделювання атаки складається з двох частин. Перша частина відповідає за отримання симульованих трас $|W|$ на основі інформації про зафіксоване розташування на кристалі елементів процесора, інформації про паразитні зв'язки на кристалі, інформації про часові параметри логічних елементів і з'єднань та інформації про внутрішню активність елементів (їх перемикання). Друга частина відповідає за аналіз $|W|$ з метою виявлення елементів ключа, які було використано пристроєм.

Розробка напівзамовлених НВІС здійснюється, як правило, системою засобів автоматизованого проектування НВІС (рис. 3).

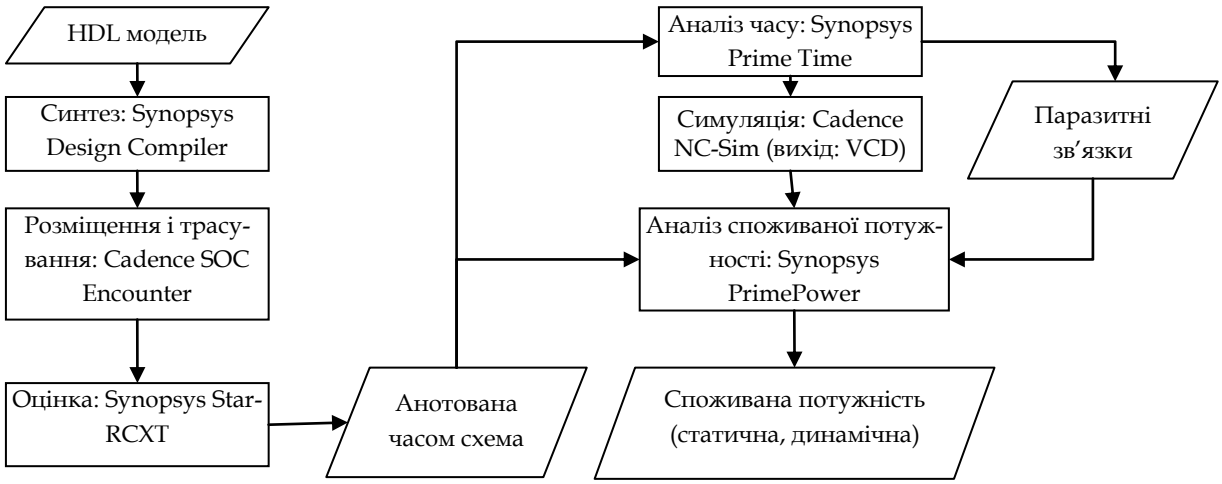


Рис. 3. Основні етапи автоматизованої розробки НВІС спеціалізованих процесорів

Спочатку створюється та відлагоджується програмна модель пристрою на одній із мов опису апаратних засобів (HDL – Hardware Description Language), наприклад Verilog, VHDL, SystemC, тощо. Далі з моделі синтезується схема пристрою, побудована на основі заданої бібліотеки елементів виробника. Після цього елементи отриманої схеми розміщуються на кристалі та проводяться з'єднання між цими елементами. Отримана інформація про розташування елементів використовується на наступному кроці для оцінки паразитних взаємовпливів елементів та з'єднань. Після цього визначаються часові затримки роботи елементів (при заданих зовнішніх факторах – температурі, напрузі живлення, тактових частотах) та отримують так звану часово-анотовану схему пристрою. Далі шляхом симулювання проводять перевірку роботи часово-анотованої схеми пристрою за допомогою системи тестів. В результаті такої симуляції отримують інформацію про внутрішню активність елементів схеми (їх перемикання, гонки) – файл VCD (Value-Change-Dump). Останнім етапом є оцінка споживаної потужності пристрою у динаміці. Для цього аналізатором споживаної потужності використовуються попередньо отримані дані про паразитні взаємовпливи елементів та з'єднань, часово-анотовану схему пристрою, файл VCD. У результаті отримують розгорнуті в часі траси споживаної потужності. Оскільки було зафіксовано остаточне розміщення елементів схеми пристрою на кристалі, для проведення DPA-атак використано лише останні два етапи розробки – симуляцію роботи пристрою за допомогою тестів та отримання трас споживаної потужності.

Система тестування пристрою написана на Verilog та складається з блоків інтерфейсу до моделі процесора, керування процесором та введення даних (рис. 4).

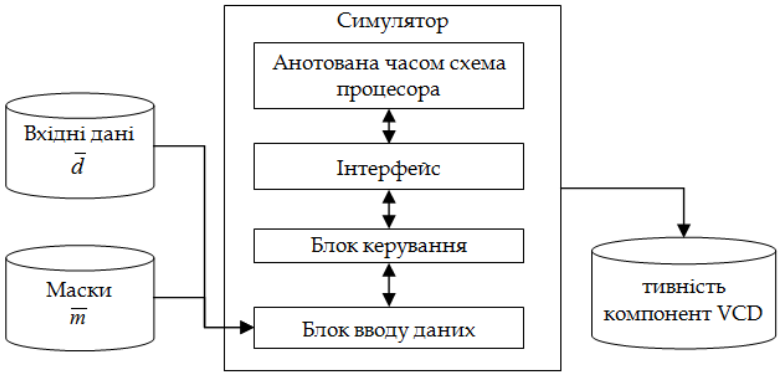


Рис. 4. Структура системи тестування процесорів шифрування даних у маскованому представленні

Вхідними даними для тестування є файли з $\bar{d}$ та попередньо згенерованими випадковими числами $\bar{m}$. Останні використано для ініціалізації та оновлення масок та симулюють результати роботи внутрішнього генератора випадкових чисел. Вихідні дані, файл VCD, формується симулятором самостійно.

Моделювання DPA атаки на HDL моделі криптографічних процесорів даних у маскованому представлені

Моделювання DPA-атаки проведено для випадків обробки даних у немаскованому та маскованому представленні. Хоча розроблені моделі процесорів обробляють дані лише у маскованому представленні, їх можна використати для обох варіантів проведення атак. Якщо встановити усі маски даних у нульове значення (усі елементи файлу $\bar{m}$ є нулями), то дані будуть оброблятися у немаскованому представленні.

При атаці процесора, який обробляє дані, згідно з алгоритмом ГОСТ 28147-89, прийнято, що усі вузли заміни є відомими. Зауважимо, що при атаці реалізацій алгоритму на програмованих процесорах, можна відновити й невідомі вузли заміни [54]. У ролі проміжного значення $f_G(d,k)$ було обрано чотири молодших біти часткового результату обчислення першого циклу:

$$f_G(d,k) = S^0(k^0_{3..0} + d_{3..0}), \quad (1)$$

де S^0 – перший вузол заміни, $k^0_{3..0}$ – чотири молодших біти першого циклового ключа, встановлено у $78_{10} = 4E_{16}$, $d_{3..0}$ – відповідні біти даних, які подають на вхід, + – операція додавання за модулем 16.

Аналогічно для процесора, який обробляє дані згідно з алгоритмом mCrypton, у ролі проміжного значення $f_C(d,k)$ було обрано чотири молодших біти проміжного результату обчислення першого циклу:

$$f_C(d,k) = S^0(K[0]_{3..0} \oplus d_{3..0}), \quad (2)$$

де S^0 – перший вузол заміни, $K[0]_{3..0}$ – чотири молодших біти першого циклового ключа, встановлено у $85_{10} = 55_{16}$, $d_{3..0}$ – відповідні біти даних, які подають на вхід, $\oplus$ – операція додавання за модулем два.

Для виявлення використаних частин циклових ключів було розроблено та реалізовано спеціальне програмне забезпечення, яким, на основі (1) і (2), було обчислено відповідні матриці $|V_G|$, $|V_C|$ усіх можливих варіантів проміжних значень на заданому наборі вхідних даних $\bar{d}_G$, $\bar{d}_C$ та елементів циклового ключа для обох алгоритмів. Після цього проведено відображення матриць $|V|$ у відповідні матриці $|H_G|$, $|H_C|$ очікуваних значень споживаної потужності із використанням спрощеної моделі на основі ваги Хемінга виду: $h_{i,j} = HW(v_{i,j}) * b$, де $b = 0.013$ – масштабний коефіцієнт вкладу однієї одиниці Хемінгової ваги даних у споживану потужність пристрою.

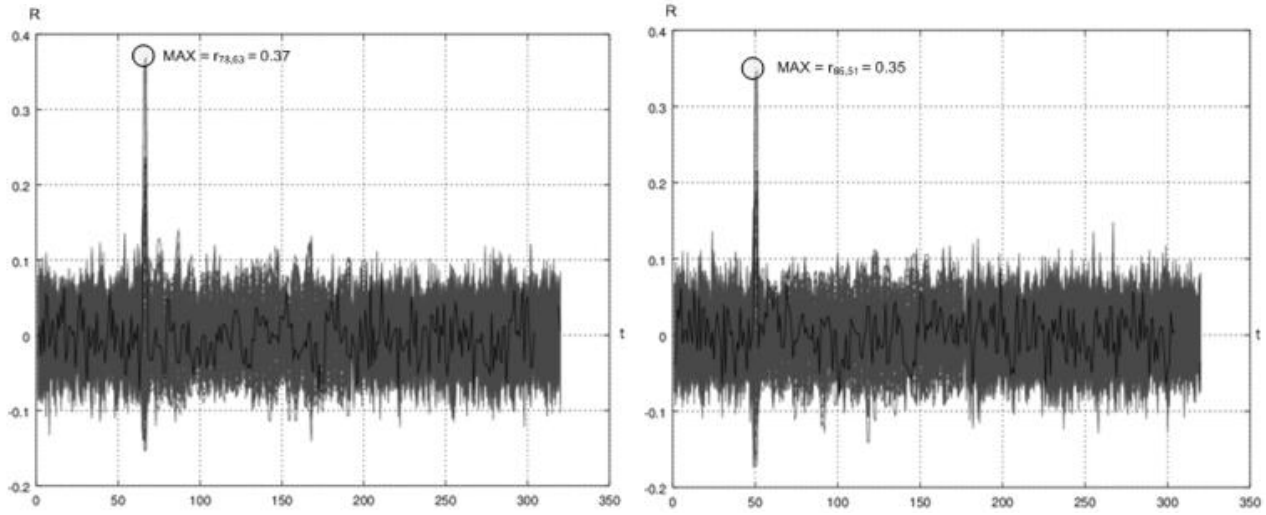
Симульовані траси споживаної потужності процесорів було отримано шляхом подавання заданих наборів вхідних даних $\bar{d}$ у симулятор, запису VCD файлів активності та подальшого їх аналізу аналізатором споживаної потужності (рис. 4). Отримані траси $|W_G|$, $|W_C|$ використано для подальшого кореляційного аналізу з метою виявлення використаних частин циклових ключів. Для цього програмне забезпечення обчислює кореляційні коефіцієнти матриці $|R|$. Подальше графічне відображення залежності отриманих коефіцієнтів від індексу передбаченого елемента циклового ключа та часового відліку дозволяє візуально визначити шуканий індекс, при якому досягається найбільша кореляція (рис. 5).

За незначного модифікування програмного забезпечення можна визначати максимальні коефіцієнти автоматично.

Як видно з рис. 5а, для процесора за ГОСТ 28147-89 найбільше значення кореляції фіксується у коефіцієнті $r_{78,63} = 0,37$, що відповідає встановленому елементу циклового ключа – 78_{10}. Аналогічно, на рис. 5б для mCrypton найбільше значення кореляції фіксується у коефіцієнті $r_{85,51} = 0,35$, що також відповідає встановленому елементу циклового ключа для цього процесора – 85_{10}. Залежність величини коефіцієнтів, які відповідають вірним елементів циклових ключів, подано темним кольором. Решта залежностей подано світлим кольором. Зауважимо, що величини кореляційних коефіцієнтів, які не відповідають встановленим елементів циклових ключів, лежать у діапа-

зоні ± 0.15, тому можуть бути легко відфільтровані навіть за візуального перегляду.

Для визначення решти елементів циклового ключа у процесорі на основі mCrypton необхідно побудувати нову атаку, цього разу взявши нове відповідне проміжне значення $f_G(d,k)$. За необхідності продовжують атаку та визначають елементи другого циклового ключа і так далі.



а) б)

Рис. 5. Результати експерименту з визначення залежності коефіцієнтів кореляції від індексу передбаченого ключа та значення траси для процесорів без маскованого представлення даних: а) для процесора за ГОСТ 28147-89, б) для процесора за mCrypton

На відміну від mCrypton, процес визначення наступних елементів циклового ключа за алгоритмом ГОСТ 28147-89 володіє такими особливостями: внаслідок використання операції додавання за модулем 2^{32}, успішність визначення кожних наступних чотирьохбітових елементів першого циклового ключа залежить від відомостей про попередні елементи. Така залежність обумовлена арифметичними переносами між розрядами при виконанні операції додавання за модулем 2^{32}. Для нівелювання впливу переносу із попередніх розрядів, можна використати відомості про вже визначені елементи циклового ключа та підібрати вхідні дані таким чином, щоб уникнути генерування такого переносу. Далі, після визначення кількох елементів циклового ключа та, за наявності обчислювальних ресурсів із достатнім об'ємом пам'яті, можна змінити проміжне значення $f_G(d,k)$ так, щоб визначити решту елементів циклового ключа методом повного перебору.

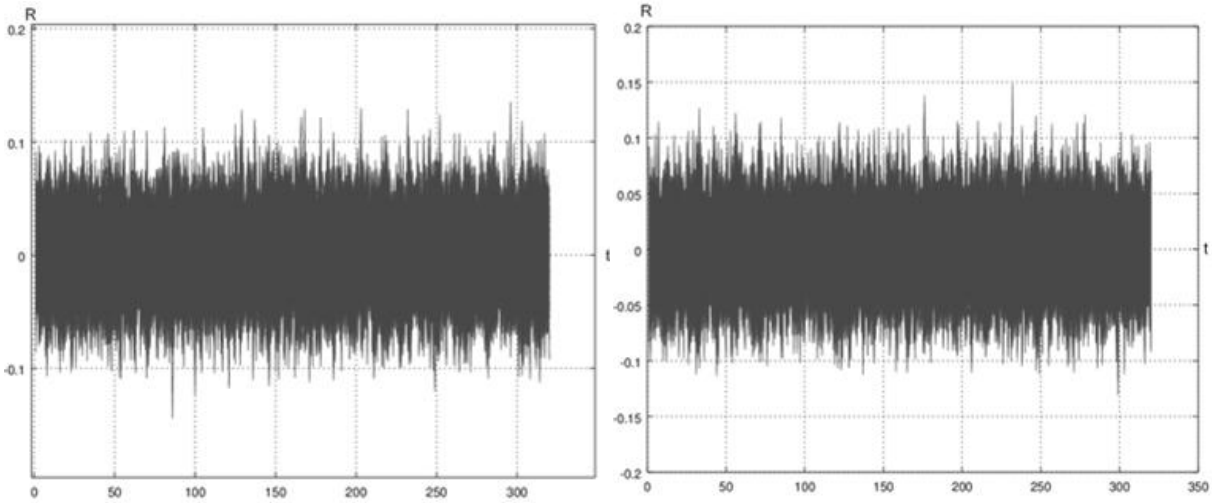
При встановленні елементів масок у відмінне від нуля значення (усі елементи файлу $\overline{m}$ є незалежними випадковими числами із рівномірним розподілом ймовірності), процесори обробляють дані у маскованому представленні із використанням цих масок. Для проведення моделювання атак було використано аналогічні функції проміжних значень. Визначення циклових ключів проводилося згідно з описаною вище методикою. У результаті обчислень матриць кореляційних коефіцієнтів $|R|$, виявлено, що значення кореляційних коефіцієнтів не виходить за межі інтервалу ± 0.15 та також відсутні їх різко виражені піки. Тому, спираючись на величини кореляційних коефіцієнтів, не вдається встановити елементи циклових ключів, оскільки ці величини для усіх варіантів елементів циклових ключів не дають однозначної відповіді про значення цих елементів ключів (рис. 6).

Таким чином, результати моделювання DPA-атаки першого порядку з використанням кореляційного аналізу на моделі процесорів, дозволяють зробити висновок про підвищену стійкість отриманих моделей процесорів до атаки такого виду.

Висновки

У роботі проведено моделювання DPA-атаки першого порядку на основі кореляційних коефіцієнтів на HDL-моделі криптографічних процесорів за алгоритмами ГОСТ 28147-89 та mCrypton, які обробляють дані у маскованому представленні. Для цього створено систему моделювання DPA-атаки, яка включає у себе аналізатор споживаної потужності, на який подають попередньо отримані дані про паразитні взаємовпливи елементів та з'єднань, часовоанотовану схему процесора, файл VCD (внутрішньої активності елементів схеми). У результаті отримують розгорнуті в часі траси споживаної потужності, які є вхідними даними для DPA-атаки першого порядку на основі кореляційних коефіцієнтів. Для такої атаки обрано проміжні значення, для яких обчислено часткові результати перших циклів алгоритмів та матриці очікуваних значень споживаної потужності із використанням спрощеної моделі на основі ваги Хеммінга. Отримані траси використано для подальшого кореляційного аналізу з метою виявлення використаних частин циклових ключів. Для цього спеціально розроблене програмне забезпечення обчислює кореляційні коефіцієнти матриці. Подальший аналіз залежності отриманих коефіцієнтів від індексу передбаченого елемента циклового ключа та часового відліку дозволяє визначити шуканий індекс, при якому досягається найбільша кореляція, тобто індекс елемента циклового ключа.

Виходячи з результатів моделювання DPA-атаки першого порядку на основі кореляційних коефіцієнтів на HDL моделі криптографічних процесорів та особливості архітектури процесорів, що полягають у обробці даних у маскованому представленні, показано, що ці процесори володіють підвищеною стійкістю до цієї атаки. Характеристики розроблених ядер процесорів дозволяють рекомендувати їх до використання у пристроях з обмеженими ресурсами (смарт-карти, криптографічні токени, мобільні пристрої зв'язку тощо), які будуть мати підвищену стійкість до DPA-атак першого порядку.



а) б)

Рис. 6. Результати експерименту з визначення залежності коефіцієнтів кореляції від індексу передбаченого ключа та значення траси для процесорів з маскованим представленням даних: а) для процесора за ГОСТ 28147-89, б) для процесора за mCrypton

Література

[1] Karpinskyy M., Korkishko L. Architecture of cryptographic devices resistant to side-channel attacks // Proc. of the International Conf. on Computer Science and Information Technologies. CSIT-2006. – Lviv: Lviv Polytechnic National University,2006. – P. 167-170.

[2] ГОСТ 28147-89. Система обработки информации. Защита криптографическая. Алгоритм криптографического преобразования. М.: Госстандарт СССР. – С. 83.

[3] Lim C.H., Korkishko T. mCrypton – a lightweight block cipher for security of low-cost RFID tags and sensors // Lecture Notes in Computer Science: Proc. of 6th International Workshop on Information Security Applications. WISA 2005. – Berlin: Springer, 2006. – Vol. 3786. – P. 243-258.

[4] Messerges T. Using second-order power analysis to attack DPA resistant software // Lecture Notes in Computer Science: Proc. of Cryptographic Hardware and Embedded Systems Workshop. CHESS-2000. – Berlin: Springer, 2000. – Vol. 1956. – P. 238-251.

[5] Mangard S., Oswald E., Popp T. Power Analysis Attacks: Revealing the Secrets of Smart Cards / Berlin: Springer, 2007. – 337 p.

[6] Agrawal D., Rao J.R., Rohatgi P. Multi-channel Attacks // Lecture Notes in Computer Science: Proc. of 5th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2003. – Cologne, Germany: Springer, 2003. – Vol. 2779. – P. 2-16.

[7] Chari S., Rao J.R. Rohatgi P. Template Attacks // Lecture Notes in Computer Science: Proc. of 4th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2002. – Redwood Shores, CA, USA: Springer, 2003. – Vol. 2523. – P. 13-28.

[8] Brier E., Clavier C., Olivier F. Correlation Power Analysis with a Leakage Model // Lecture Notes in Computer Science: Proc. of 6th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2004. – Cambridge, MA, USA: Springer, 2004. – Vol. 3156. – P. 16-29.

[9] Kocher P., Jaffe J., Jun B. Differential power analysis // Lecture Notes in Computer Science: Proc. of International Conf. Advances in Cryptology. CRYPTO-1999. – Berlin: Springer, 1999. – Vol. 1666. – P. 388-397.

[10] Irwin J., Page D., Smart N. Instruction Stream Mutation for Non-Deterministic Processors // IEEE Computer Society: Proc. of IEEE International Conference on Application-Specific Systems, Architectures and Processors. – 2002. – P. 286-295.

[11] Yang S., Wolf W., Vijaykrishnan N., Serpanos D., Xie Y. Power Attack Resistant Cryptosystem Design: A Dynamic Voltage and Frequency Switching Approach // IEEE Computer Society: Proc. of Design, Automation and Test in Europe Conference and Exposition. DATE-2005. – Munich, 2005. – P. 64-69.

[12] Karpinskyy M., Korkishko L., Korkishko T. Randomized execution of regular cryptographic algorithms // Proc. of 3-rd International Conf. «Advanced Computer Systems and Networks: Design and Application» (ACSN'2007). –Lviv, 2007. – P. 114-117.

[13] May D., Muller H. L., Smart N. P. Non-deterministic Processors // Lecture Notes in Computer Science: Proc. of 6th Australasian Conference Information Security and Privacy. ACISP-2001. – Berlin: Springer, 2001. – Vol. 2119. – P. 115-129.

[14] Clavier C., Coron J., Dabbous N. Differential Power Analysis in the Presence of Hardware Countermeasures // Lecture Notes in Computer Science: Proc. of Second International Workshop Cryptographic Hardware and Embedded Systems. CHES-2000. – Worcester, MA, USA: Springer, 2000. – Vol. 1965. – P. 252-263.

[15] Ratanpal G.B., Williams R.D., Blalock T.N. An On-Chip Signal Suppression Countermeasure to Power Analysis Attack // IEEE Transactions on Dependable and Secure Computing. – 2004. – Vol. 1(3). – P. 179-189.

[16] Muresan R., Vahedi H., Zhanrong Y., Gregori S. Power-Smart System-On-Chip Architecture for Embedded Cryptosystems // Proc. of the 3rh IEEE/ACM/IFIP International Conf. on Hardware/Software Codesign and System Synthesis. – ACM Press, 2005. – P. 184-189.

[17] Mesquita D., Techer J-D., Torres L., Sassatelli G., Gambon G., Robert M., Moraes F. Current Mask Generation: A Tansistor Level Security Against DPA Attack. // Proc. of the 18th Annual Symposium on Integrated Circuits and System Design SBCCI'05. – ACM Press, 2005. – P. 115-120.

[18] Benini L., Macii A., Macii E., Omerbegovic E., Pro F., Poncino M. Energe-Aware Design Techniques for Differential Power Analysis Protection // Proc. of 40th Design, Automation Conf., DAC-2003. – ACM Press, 2003. – P. 36-41.

[19] Benini L., Macii A., Macii E., Omerbegovic E., Poncino M., Pro F. A Novel Architecture for Power Maskable Arithmetic Units // Proc. of 13th ACM Great Lakes Symposium on VLSI 2004. – Washington: ACM Press, 2003. – P. 136-140.

[20] Tiri K., Akmal M., Verbauwhere I. A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards // Proc. of IEEE 28th Europpean Solid-State Circuits Conf. ESSCIRC-2002. – Florence, 2002. – P. 403-406.

[21] Tiri K., Verbauwhere I. A Logic Level Design Methodology for Secure DPA Resistant ASIC or FPGA Imlementation // IEEE Computer Society: Proc. of 2004 Design, Automation and Test in Europe Conference and Exposition. DATE-2004. – Paris, 2004. – Vol. 1. – P. 246-251.

[22] Bystrov A., Sokolov D., Yakovlev A., Koelmans A. Balancing Power Sinature in Secure Systems // Proc. of 14th UK Asynchronous Forum. – Newcastle, 2003. – [Цит. 2003, 12 червня]. – Режим доступу: < http://www.staff.ncl.ac.uk/i.g.clark/async/ukasyncform14/forum14-papers/forum-bystrov.pdf>.

[23] Sokolov D., Murphy J., Bystrov A., Yakovlev A. Improving the Security of Dual-Rail Circuits // Lecture Notes in Computer Science: Proc. of 6th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2004. – Berlin:Springer, 2004. – Vol. 3156. – P. 282-297.

[24] Sokolov D., Murphy J., Bystrov A., Yakovlev A. Design and Analysis of Dual-Rail Circuits for Security Applications // IEEE Transactions on Computers, 2005. – Vol. 54(4). – P. 449-460.

[25] Bucci M., Giancane L., Luzzi R., Trifiletti A. Three-Rhase Dual-Rail Pre-Charge Logic // Lecture Notes in Computer Science: Proc. of 8th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2006. – Berlin:Springer, 2006. – Vol. 4249. – P. 282-297.

[26] Aigner M., Mangart S., Menicocci R., Olivieri M., Scotti G., Trifiletti A. A Novel CMOS Logic Style with Data Independent Power Consumotion // Proc. of IEEE Inernational Symposium on Circuits and Systems. ISCAS-2005. – 2005. – Vol. 2. – P. 1066-1069.

[27] Coron J-S., Kocher P.C., Naccache D. Statistics and Secret Leakage // Lecture Notes in Computer Science: Proc. of 4th International Conference Financial Cryptography. FC-2000. – Berlin:Springer, 2001. – Vol. 1962. – P. 157-173.

[28] Shamir A. Protection Smart Cards from Passiv Power Analysis with Detachted Power Supplies // Lecture Notes in Computer Science: Proc. of Second International Workshop Cryptographic Hardware and Embedded Systems. CHES-2000. – Berlin:Springer, 2000. – Vol. 1956. – P. 71-77.

[29] Corsonello P., Perri S., Margala M. A New Charge-Pump Based Countermeasure Against Differential Power Analysis // Proc. of the 6th International Conference on ASIC. ASICON-2005. – IEEE, 2005.– Vol. 1. – P. 66-69.

[30] Moore S., Anderson R.J., Cunningham P., Mullins R.D., Taylor G.S. Improving Smart Card Security using Self-timed Circuits // Proc. of Eighth Inernational Symposium on Asynchronous Circuits and Systems. ASYNC-2002. – IEEE Computer Society, 2002. – P. 211-218.

[31] Yu Z.C., Furber S.B., Plana L.A. An Investigation into the Security of Self-Timed Circuits // Proc. of 9th Inernational Symposium on Advanced Research in Asynchronous Circuits and Systems. ASYNC-2003. – IEEE Computer Society, 2003. – P. 206-215.

[32] Kulikowski K.J., Su M., Smirnov A. B., Taubin A., Karpovsky M.G., MacDonald D. Delay Insensitive Encoding and Power Analysis: A Balancing Act // In 11th Inernational Symposium on Advanced Research in Asynchronous Circuits and Systems. ASYNC 2005. – IEEE Computer Society, 2005. – P. 116-125.

[33] Kulikowski K.J., Smirnov A. B., Taubin A. Automated Design of Chryptographic Devices Resistant to Multiple Side-Channel Attacks // Lecture Notes in Computer Science: Proc. of 8th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2006. – Berlin: Springer, 2006. – Vol. 4249. – P. 399-413.

[34] Yu A., Bree D.S. A Clock-less Implementation of the AES Resists to Power and Timing Attacks // Proc. of International Conf. on Information Technology: Coding and Coputing. ITCC-2004. – IEEE Computer Society, 2004. –Vol. 2. – P. 525-532.

[35] Goubin L., Patarin J. DES and Differential Power Analysis – The Duplication Method // Lecture Notes in Computer Science: Proc. of First International Workshop Cryptographic Hardware and Embedded Systems. CHES-1999. – Berlin:Springer, 1999.– Vol. 1717. – P. 158-172.

[36] Chari S., Jutla C.S., Rao J.R., Rohatgi P. A Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards // Proc. of Second Advanced Encryption Standart (AES) Candidate Conference. – Roma, 1999.

[37] Messerges, T. S. Securing the AES finalists against power analysis attacks // Lecture Notes in

Computer Science: Proc. of Workshop Fast Software Encryption. – Berlin: Springer, 2000. – Vol. 1978. – P. 150-165.

[38] Popp T., Mangard S. Masked Dual-Rail Pre-Charge Logic: DPA-Resistance without Routing Constraints // Lecture Notes in Computer Science: Proc. of 7th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2005. – Berlin:Springer, 2005. – Vol. 3659. – P. 172-186.

[39] Popp T., Mangard S. Implementation Aspects of the DPA- Resistant Logic Style MDPL // Proc. of Inernational Symposium on Circuits and Systems. ISCAS- 2006. – IEEE, 2006. – P. 2913-2916.

[40] Suzuki D., Saeki M., Ichikawa T. Random Switching Logic: A Countermeasure against DPA based on Transition Probability // Cryptogy ePrint Archive (http://eprint.iacr.org/), Report 2004/346, 2004.

[41] Chen Z., Zhou Y. Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage // Lecture Notes in Computer Science: Proc. of 8th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2006. – Berlin:Springer, 2006. – Vol. 4249. – P. 242-254.

[42] Trichina E., Korkishko T., Lee K-H. Small Size, Low Power, Side Channel-Immune AES Coprocessor: Design and Synthesis Results // Lecture Notes in Computer Science: Proc. of 4th Conference Advanced Encryption Standart. AES-2004. – Berlin: Springer, 2005. – Vol. 3373. – P. 113-127.

[43] Golic J.D., Menicocci R. Universal Masking on Logic Gate Level // IEE Electronic Letters. – 2004. – Vol. 40(9). – P. 526-527.

[44] Ishai Y., Sahai A., Wagner D. Private Circuits: Securing Hardware against Probing Attacks // Lecture Notes in Computer Science: Proc. of 23th Annual Inernational Cryptologe Conference Advances in Cryptology. CRYPTO-2003. – Berlin: Springer, 2003. – Vol. 2729. – P. 463-481.

[45] Коркішко Л. Операція множення даних у маскованому представленні // Матеріали XI наукової конференції Тернопільського державного технічного університету ім. І.Пулюя. – Тернопіль, 2007. – С. 83.

[46] Benini L., Galati A., Macii A., Macii E., Poncino M. Energi-Efficient Data Scrambling on Memory-Prosessor Interfaces // Proc. of International Symposium on Low Power Electronics and Desin. – Berlin: Springer, 2003. – P. P. 26-29.

[47] Golik J. D. DeKaRT: A New Paradigm for Key-Dependent Reversible Circuits // Lecture Notes in Computer Science: Proc. of 5th International Workshop Cryptographic Hardware and Embedded Systems. CHES-2003. – Berlin: Springer, 2003. – Vol. 2779. – P. 98-112.

[48] Elbaz R., Torres L., Sassatelli G., Guillemin P., Anguille C., Bardouillet M., Buatois C., Rigaud J-B. Hardware Engines for Bus Encryption: A Survey of Existing Techniques // Proc. of Design, Automation and Test in Europe Conference and Exposition. DATE-2005. – IEEE Computer Society, 2005. – P. 40-45.

[49] Bucci M., Gugieimo M., Luzzi R., Trifiletti A. A Power Consumption Randomization Countermeasure for DPA-Resistant Cryptographic Processors // Lecture Notes in Computer Science: Proc. of 14th International Workshop on Integrated Circuit and System Design, Power and Timing Modeling, Optimization and Simulation. PATMOS 2004. – Berlin: Springer, 2004. – Vol. 3254. – P. 481-490.

[50] Карпінський М.П., Коркішко Л.М., Коркішко Т.А. Адаптування алгоритмів криптографічних перетворень до обробки маскованих даних// Вісник хмельницького національного університету – 2007. – №3, Том 1 – С. 67-70.

[51] Карпінський М.П., Коркішко Л.М. Процесор симетричного блокового шифрування за ГОСТ 28147-89 для даних у маскованому представленні // Матеріали 2-ї міжнародної конференції «Комп'ютерні науки та інженерія» (CSE`2007). – Львів, 2007. – С. 86-90.

[52] Карпінський М.П., Коркішко Л.М. Захист двійкових суматорів від інженерно-криптографічних атак за побічними каналами витоку інформації // Матеріали 1-ї міжнародної конференції «Комп'ютерні науки та інженерія» (CSE`2006). – Львів, 2006. – С. 58-61.

[53] Karpinskyy M., Korkishko L., Furmanyuk A. Masked Encryption Algorithm mCrypton for Resource-Constrained Devices // Proc. of 4th International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2007). – Dortmund, 2007. – P. 628-633.

[54] Карпінський М., Коркішко Л., Коркішко Т. Інженерно-криптографічна атака за аналізом споживаної потужності на програмно-апаратні реалізації криптографічного перетворення за чинним стандартом // Вісник Тернопільського державного технічного університету. – 2005. – №3. – С. 127-135.

УДК 004.056.5 (045)

Карпинский М.П., Коркишко Л. М. Моделирование DPA атаки первого порядка

Аннотация. *В данной работе проведено моделирование DPA-атаки первого порядка на основе корреляционных коэффициентов на HDL-модели криптографических процессоров за алгоритмами ГОСТ 28147-89 и mCrypton, которые обрабатывают данные в маскированном представлении. Для этого создано систему моделирования DPA-атаки, которая включает в себя анализатор потребляемой мощности, на который подают предварительно полученные данные о паразитных взаимовлияниях элементов и соединений, временно-аннотированую схему процессора, файл VCD (внутренней активности элементов схемы). В результате получают развернутые во времени трасы потребляемой мощности, которые являются входными данными для DPA-атаки первого порядка на основе корреляционных коэффициентов. Выходя из результатов моделирования этой HDL-модели ядер криптографических процессоров и особенности архитектуры рассмотренных процессоров, показано, что эти процессоры владеют повышенной стойкостью к этой атаке. Характеристики разработанных ядер процессоров позволяют рекомендовать их к использованию в устройствах с ограниченными ресурсами (смарт-карты,*

криптографические токены, мобильные устройства связи и т.д.), которые будут иметь повышенную стойкость к DPA-атакам первого порядка.
Ключевые слова: *защита информации, DPA атаки, ядра процессоров шифрования, маскированное представление данных, канал утечки информации.*

Karpinski M., Korkishko L. DAP attack modelling of first degree
Abstract. *This work present first order DPA attack based on correlation coefficients on HDL models of cryptographic processors using symmetric ciphers GOST 28147-89 and mCrypton and processing data in masked representation. A system for DPA attack modeling was created, including power consumption analyzer, processing data about layout parasitic interconnections of elements and connections, time-annotated post-place-and-rout information, processor internal elements activity data. As the result of the analysis, we obtained power consumption traces, serving as input for the first order DPA attack based on correlation coefficients. Based on modeling results of the attack on HDL models of the cryptographic processors and architecture features of the processors, we conclude that used processors cores have increased resistance to the attack. As the result, we can recommend the cores of the processors to be used in resource-constrained devices (smart-cards, cryptographic tokens, mobile devices) with higher resistance to first order DPA attack.*
Key words: *information security, DPA attacks, cores of cryptographic processors, masked data representation, information leakage by side-channel.*

Отримано *29 квітня 2016 року*, затверджено редколегією *13 травня 2016 року*