УДК 621.3

Коркішко Л.М.

ОПЕРАЦІЙНІ ПРИСТРОЇ ЛОГІЧНИХ ОПЕРАЦІЙ НАД ДАНИМИ У МАСКОВАНОМУ ПРЕДСТАВЛЕННІ

Тернопільський національний економічний університет

Запропоновано структури операційних пристроїв логічних операцій над даними у маскованому представлені. До переліку логічних операцій входять операції логічного множення та додавання. Досліджено характеристики складності цих структур

Вступ

Зростання цінності інформації, яка зберігається, обробляється та передається у комп'ютерних системах зумовило зростання актуальності задач забезпечення конфіденційності, цілісності та автентичності інформації при зростанні ймовірності реалізації загроз несанкціонованого доступу до такої інформації. У комп'ютерних системах перелічені задачі часто вирішуть шляхом криптографічних перетворень інформації. Актуальним напрямком визначення параметрів криптографічних перетворень є аналіз залежності спостережуваних фізичних характеристик комп'ютерних пристроїв які їх реалізують (час обробки, споживаний струм, електромагнітне випромінювання, тощо) від даних, які обробляються – так звані "інженерно-криптографічні атаки" [1–4].

На даний час відомі рішення, які дозволяють будувати комп'ютерні компоненти на базі спеціальної елементної бази (логічних елементах), яка враховує можливі канали витоку інформації, зокрема через енергоспоживання (споживану потужність). Проте коштовність масового виробництва таких пристроїв стримує їх впровадження та використання. Альтернативні методи протидій таким атакам основані на спеціальному представленні даних при їх обробці та зберіганні [5–8]. Так зване "масковане представлення" (техніки розділення таємниці) при побудові арифметичних та логічних методів обробки даних, дозволяє будувати комп'ютерні компоненти на базі традиційної елементної бази, які володіють підвищеною стійкістю до атак на основі енергоспоживання та низькою вартістю виготовлення.

Отже, створення нових методів та структур операційних пристроїв арифметичної та логічної обробки даних у маскованому представленні, які дозволяють будувати комп'ютерні компоненти з підвищеною стійкістю до атак на основі енергоспоживання та низькою вартістю виготовлення на базі стандартизованих логічних елементів є актуальною науково-технічною задачею.

Логічні операції над даними у маскованому представленні

Для створення методу виконання двомісних логічних операцій Булевої алгебри над даними у маскованому представленні, скористаємося алгебраїчним методом, запропонованим у [9]. Згаданий метод використовує дистрибутивні властивості операцій логічного множення та додавання за модулем 2 та дозволяє виконання операції логічного множення над даними у маскованому представленні з використанням логічної маски. Для виконання операцій над даними з більшою кількістю масок (дві та більше) запропоновано параметризований алгебраїчний метод виконання логічних операцій (логічного множення і додавання), де параметром є кількість використаних логічних масок [10].

Припустимо, що a і b є немаскованими даними, над якими необхідно виконати двомісну логічну операцію у маскованому представленні з використанням заданої кількості масок. Нехай дані a і b подані у маскованому представленні з використанням логічного маскування: $\tilde{a} = a \oplus x_1 \oplus \ldots \oplus x_n$, і $\tilde{b} = b \oplus y_1 \oplus \ldots \oplus y_n$, де $x = x_1, \ldots, x_n$, $y = y_1, \ldots, y_n$ – маски відповідно a і b, що є незалежними випадковими числами з рівномірним розподілом ймовірності. Задача обчислення логічної операції над даними у маскованому пред-

ставленні формулюється так: за заданими наборами $\{\tilde{a},x\}$, $\{\tilde{b},y\}$ обчислити без розголошення відомостей про a і b результат $\{\tilde{c},z\}$, де $\tilde{c}=(a*b)\oplus z_1\oplus\ldots\oplus z_n$, "$*$" – позначення логічної операції, $z=z_1,\ldots,z_n$ – маски результату, що є незалежними випадковими числами з рівномірним розподілом ймовірності.

Використовуючи властивості операцій логічного множення, логічного додавання і додавання за модулем два, автором запропоновано методи виконання логічних операцій множення та додаваннянад даними, поданими у маскованому представленні з використанням n логічних масок, відповідно [10]:

$$\tilde{c}=a\cdot b\oplus\bigoplus_{i=1}^{n}z_i=\tilde{a}\cdot\tilde{b}\oplus\bigoplus_{i=1}^{n}x_i\cdot\tilde{b}\oplus\bigoplus_{j=1}^{n}y_j\cdot\tilde{a}\oplus\bigoplus_{\substack{i=1\\j=1}}^{n}x_i\cdot y_j\oplus\bigoplus_{i=1}^{n}z_i, \quad (1)$$

$$\tilde{c}=(a\vee b)\oplus\bigoplus_{i=1}^{n}z_i=\tilde{a}\vee\tilde{b}\oplus\bigoplus_{i=1}^{n}x_i\cdot\tilde{b}\oplus\bigoplus_{j=1}^{n}y_j\cdot\tilde{a}\oplus\bigoplus_{\substack{i=1\\j=1}}^{n}x_i\cdot y_j\oplus\bigoplus_{i=1}^{n}x_i\oplus\bigoplus_{j=1}^{n}y_j\oplus\bigoplus_{i=1}^{n}z_i, \quad (2)$$

де "$\cdot$" – операція логічного множення, "$\vee$" – операція логічного додавання.

Особливістю цих методів є уникнення обчислень із відкритими даними a і b. При цьому можна запропонувати декілька варіантів виконання виразів (1) та (2), які будуть відрізнятися між собою порядком виконання операцій. Вираз для обчислення логічного множення даних у маскованому представленні із використанням однієї маски описано у [9] та є частковим випадком виразу (1) при $n=1$.

Отже, використовуючи вирази (1) і (2), можна створити базові логічні операційні пристрої для подальшої побудови на їх основі комп'ютерних компонентів для виконання крипто-графічних перетворень, які є дозволяють обробляти дані, подані у маскованому з використанням n логічних масок. При цьому розроблені методи дозволяють будувати необхідні обчислення за заданим n, яке використовуються для подання даних і результатів.

Структури операційних пристроїв логічних операцій

Для побудови структури операційного пристрою логічного множення двох даних a і b, поданих у маскованому представленні виду $\{\tilde{a},x\}$, скористаємось виразом (1). Для цього перепишемо цей вираз у більш зручній формі

$$\tilde{c}=\tilde{a}\cdot\tilde{b}\oplus\bigoplus_{i=1}^{n}[(\tilde{a},\tilde{b})\cdot(y_i,x_i)^T\oplus(x_1,\ldots,x_n)\cdot(ROT_{i-1}(y_1,\ldots,y_n))^T\oplus z_i], \quad (3)$$

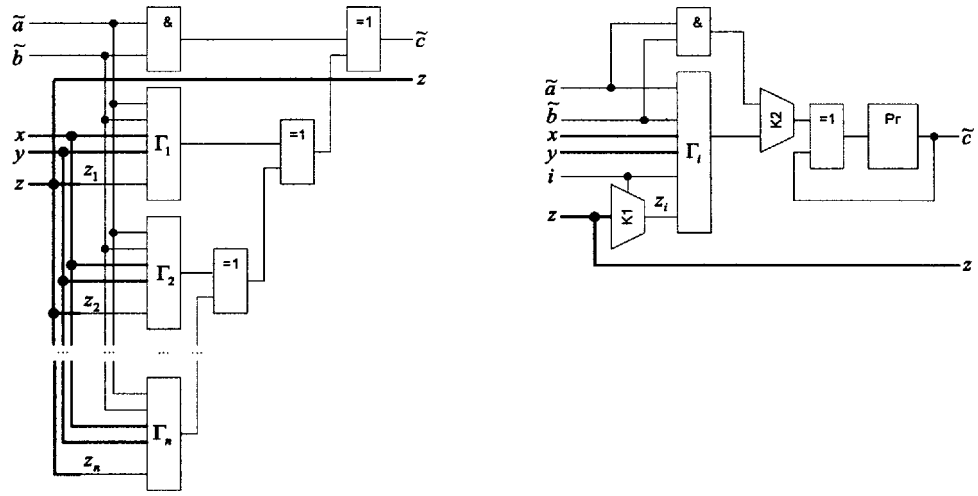
де $ROT_{i-1}(y)$ – функція циклічного зсуву вектора y на i позицій ліворуч, множення векторів відбувається з використанням операцій логічного множення, а додавання множників – за допомогою операції додавання за модулем два.

У виразі (3) виділимо "групові" доданки виду

$$\Gamma_i=(\tilde{a},\tilde{b})\cdot(y_i,x_i)^T\oplus(x_1,\ldots,x_n)\cdot(ROT_{i-1}(y_1,\ldots,y_n))^T\oplus z_i. \quad (4)$$

Використовуючи метод прямого апаратного відображення потокового графу [11] виразу (3) з врахуванням (4), структура операційного пристрою для виконання операцій логічного множення буде мати вид рис. 1а. Структура операційного пристрою (рис. 1а) містить n однакових блоків Γ_i, кожен з яких виконує операції згідно з (4). Операції обчислення термів логічного множення виконуються паралельно, а операції додавання за модулем два термів – послідовно.

Зауважимо, що доданки виду (4) є базовими операціями для логічного множення даних у маскованому представленні з використанням n масок, які відрізняються лише порядком використання вхідних даних та кількістю позицій циклічного зсуву ліворуч, заданих індексом i. Тому, можна запропонувати іншу структуру операційного пристрою (рис. 1б), базовану на апаратному відображенні згортки потокового графу [9], що заданий виразом (3).



а) б)

Рис. 1. Структури операційного пристрою для виконання операцій логічного множення над даними у маскованому представленні з використанням n масок: а) на базі прямого апаратного відображення потокового графу, б) на базі апаратного відображення згортки потокового графу

Розгортка потокового графу у часі здійснюється шляхом зміни індексу i, керування регістром Рг та керування адресними входами комутаторів К1 і К2. Спочатку в регістр Рг записується нульове значення. Далі адресний вхід комутатора К2 встановлюється таким чином, щоб на вхід елемента додавання за модулем два подавався вихід блоку Γ_i. Подальше встановлення значень індексу i від 1 до n та записування даних у Рг дозволяє обчислити суму $\bigoplus_{i=1}^{n}\Gamma_i$. Для завершення обчислення виразу (3) комутатор К2 подає на вхід суматора результат логічного множення маскованих даних. Після запису результату у Рг, регістр буде містити результат обчислення виразу (3).

Для створення структури операційного пристрою логічного додавання двох даних a і b, поданих у аналогічному маскованому представленні, скористаємось виразом (2). Для цього, врахувавши вираз (4) та увівши позначення

$$\Gamma_i' = \Gamma_i \oplus x_i \oplus y_i, \qquad (5)$$

перепишемо вираз (2) у зручній формі:

$$\tilde{c} = \tilde{a} \vee \tilde{b} \oplus \bigoplus_{i=1}^{n}\Gamma_i'. \qquad (6)$$

Структура операційного пристрою для виконання операцій логічного додавання буде повторювати структуру операційного пристрою на рис. 1а. Відмінними будуть: елемент логічного додавання замість елемента логічного множення та блоки, які реалізує функцію Γ_i' згідно з (5) замість блоків Γ_i. Аналогічні зміни відбудуться й у пристрої із структурою на рис. 1б.

Вибір структури операційного пристрою залежить від заданих розробнику обмежень на обсяг використаного обладнання, час виконання операції та кількості масок у маскованому представленні даних.

Дослідження характеристик запропонованих операційних пристроїв

До оцінки характеристик операційних пристроїв дослідимо такі характеристики складності їх структур [12]: апаратну, часову і місткістну складність. Додатково оцінимо розмір вибірки випадкових чисел, необхідних для виконання обчислень.

Апаратна складність оцінюється як кількість умовних (типових) елементів, необхідних для побудови операційного пристрою. Часова складність оцінюється як довжина критичного шляху виконання того чи іншого процесу. При апаратній реалізації операційного пристрою часова складність відображає максимальну затримку обробки даних при заданому порядку виконанні обчислень. При програмному виконанні – загальний час, який не-

обхідно витратити на обробку даних. Місткістна складність визначається як розмір пам'яті, необхідний для зберігання проміжних результатів обчислень.

Для оцінки характеристик операційних пристроїв виконання логічних операцій над даними у маскованому представленні скористаємося виразами (3) і (6). Приймемо, що порядок виконання складових операцій цих виразів не впливає на характеристики апаратної складності.

Для виразів (3) і (6) критичний шлях визначається порядком виконання обчислень. Порядок виконання обчислень впливає на рівень витоку інформації із операційних пристроїв. Для цього у виразі (3) обчислення доданків $\tilde{a}\cdot\tilde{b}$, $x_i\cdot\tilde{b}$, $y_j\cdot\tilde{a}$ та $x_i\cdot y_j$ можна проводити паралельно, а маска результату вводиться поступово, починаючи з $x_i\cdot y_j$ і закінчуючи $\tilde{a}\cdot\tilde{b}$. Тому, часова складність виконання виразу (3) t_{MAND} для обробки даних у маскованому представленні із використанням $n>1$ масок буде залежати від тривалості виконання однієї операції логічного множення однобітових даних $t_\wedge$, додавання за модулем 2 однобітових даних $t_\oplus$ і становитиме $t_{MAND}(n)=3nt_\oplus+n^2t_\oplus$, а часова складність виконання виразу (3) для захисту обробки даних з однією маскою, тобто $n=1$, дорівнюватиме $t_{MAND}=t_\wedge+4t_\oplus$.

Аналогічно, для виразу (6) обчислення доданків $\tilde{a}\vee\tilde{b}$, $x_i\cdot\tilde{b}$, $y_j\cdot\tilde{a}$, $x_i\cdot y_j$ можна проводити паралельно, а маску результату вводити поступово, починаючи з x_i, y_j і закінчуючи $\tilde{a}\vee\tilde{b}$. Часова складність виконання виразу (6) для обробки даних у маскованому представленні із використанням n масок t_{MOR} буде залежати від тривалості виконання однієї операції логічного додавання однобітових даних $t_\vee$, $t_\wedge$ та $t_\oplus$ і складе $t_{MOR}(n)=5nt_\oplus+n^2t_\oplus$.

Паралельне виконання виразів (3) і (6) можна використати для апаратної реалізації операційних пристроїв логічних операцій над даними у маскованому представленні. Для побудови графіків залежності часової складності виконання маскованих операцій від кількості масок припустимо, що $t_\wedge\approx t_\oplus$ (рис. 2а).

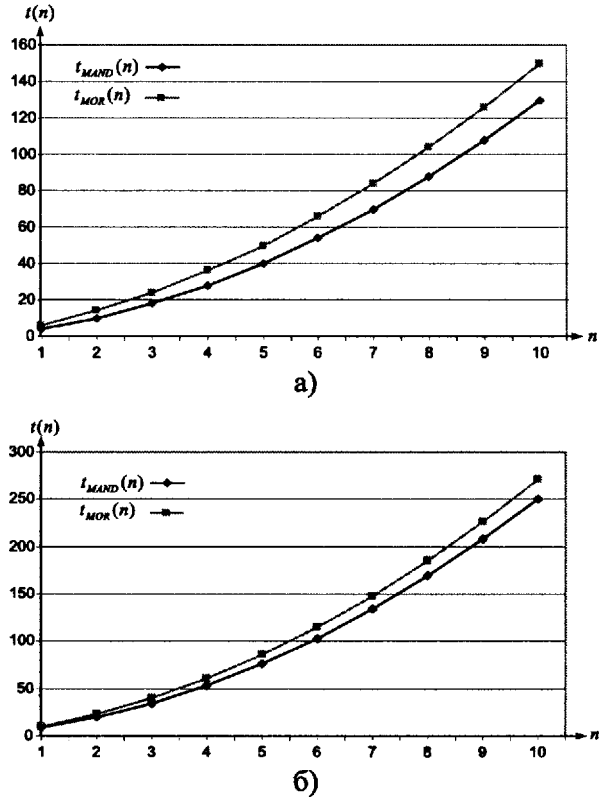


Рис. 2. Графіки залежності часової складності операційних пристроїв від кількості масок: а) паралельне виконання, б) послідовне виконання

З наведених на рис. 2а графіків випливає, що при використанні n масок час виконання маскованих операцій буде зростати пропорційно до n^2. При цьому, за однакових n часова складність виконання операції логічного множення над маскованими даними є меншою за відповідну часову складність для виконання операції логічного додавання.

При альтернативному виконанні усіх операцій послідовно (у заданому порядку) згідно з (3), часова складність виконання маскованої операції логічного множення становитиме

$t_{MAND}(n)=t_\wedge+nt_\oplus+(2n+n^2)(t_\oplus+t_\wedge)$. Аналогічно, для виразу (6) отримаємо $t_{MOR}(n)=t_\vee+3nt_\oplus+(2n+n^2)(t_\oplus+t_\wedge)$.

Послідовне виконання виразів (3) і (6) можна використати при програмній реалізації обчислень над маскованими даними на універсальних програмованих

процесорах. Оскільки, у цих процесорах час виконання інструкцій логічних операцій є приблизно однаковий і майже не залежить від типу цих логічних операцій, то для побудови графіків залежності часової складності виконання операцій над маскованими даними приймемо, що $t_{\wedge} \approx t_{\oplus} \approx t_{\vee}$. З наведених на рис. 2б графіків випливає, що при використанні наведених на рис. 1а структур операційних пристроїв час виконання маскованих операцій буде змінюватися аналогічно до складності паралельного способу виконання – пропорційно до квадрату кількості використаних масок. При цьому, за однакової кількості масок часова складність виконання операції логічного множення над даними у маскованому представленні є меншою за відповідну часову складність для виконання операції логічного додавання.

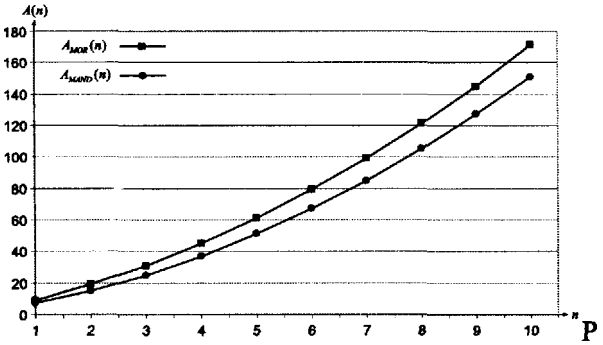
Оцінимо апаратну складність операційних пристроїв, структури яких наведено на рис. 1а через кількість двовходових логічних елементів логічного множення $N_{\wedge}$, логічного додавання $N_{\vee}$, додавання за модулем 2 $N_{\oplus}$ (табл. 1).

Результат аналізу графіків (рис. 3) залежності апаратної складності цих структур свідчить, що при використанні n для маскованого представлення даних апаратна складність операційних пристроїв буде зростати пропорційно до n^2.

Таблиця 1. Залежність апаратної складності A розроблених операційних пристроїв від кількості масок n

Логічна операція	$N_{\vee}$	$N_{\wedge}$	$N_{\oplus}$
Множення	0	$1+2n+n^2$	$3n+n^2$
Додавання	1	$2n+n^2$	$5n+n^2$

Графіки рис. 3 отримано з врахуванням, що $A_{\oplus} = 3A_{\wedge}$, де $A_{\oplus}(n) = 2n+1$. Розмір вибірки випадкових чисел для операційних пристроїв логічних операцій над маскованими даними є рівним кількості масок, які використано для представлення вхідних даних – n.



ис. 3. Графік залежності апаратної складності операційних пристроїв від кількості масок

У порівнянні з відомими структурами операційних пристроїв операцій логічного множення і додавання над даними у маскованому представленні [13], побудовані автором операційні пристрої володіють на 33% меншою часовою складністю при послідовному виконанні операцій.

Висновки

У роботі запропоновано структури операційних пристроїв логічних операцій над даними у маскованому представленні. Розроблені структури дозволяють виконувати операції логічного множення та додавання та є параметризованими до кількості використаних масок у маскованому представленні. Дослідження запропонаваних структур показало, що їх апаратна складність збільшується як квадрат кількості використаних масок. Аналогічно змінюється часова складність. При цьому, за однакових n апаратна складність операційного пристрою логічного множення над маскованими даними є меншою за відповідну апаратну складність операційного пристрою операції логічного додавання.

Отримані результати можна використати при проектуванні комп'ютерних компонентів систем захисту інформації, у яких є можливість реалізації загрози отримання інформації про параметри криптографічних перетворень шляхом проведення інженерно-криптографічних атак.

Список літератури

1. *Zhou Y. B., Feng D. G.* Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing // Proc. of National Institute of Standardization Physical Security Testing Workshop. – 2006.

– [Цит. 2006, 10 січня] – Доступний з <http://csrc.nist. gov /cryptval/ physec/ papers/ physecpaper19. pdf>.

2. *Anderson R.J., Kuhn M.G.* Tamper Resistance – a Cautionary Note // Proc. of The Second USENIX Workshop on Electronic Commerce. – Oakland, 1996. – P. 1 – 11.

3. *Skorobogatov S.* Semi-Invasive Attacks – A New Approach to Hardware Security Analysis / University of Cambridge. – Cambridge, 2004. – 196 p.

4. *Kocher P.* Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems // Lecture Notes in Computer Science: Proc. of International Conf. Advances in Cryptology. CRYPTO-1996. – Berlin: Springer, 1996. – Vol. 1109. – P. 104 – 113.

5. *Kocher P., Jaffe J., Jun B.* Using unpredictable information to minimize leakage from smartcards and other cryptosystems // USA Patent, International Publication. – 1999. – WO 99/63696.

6. *Chari S., Jutla C. S., Rao J.R., Rohatgi P.* Towards sound approaches to counteract power analysis attacks // Lecture Notes in Computer Science: Proc. of International Conf. Advances in Cryptology. CRYPTO-1999. – Berlin: Springer, 1999. – Vol. 1666. – P. 398 – 412.

7. *Mangard S., Oswald E., Popp T.* Power Analysis Attacks: Revealing the Secrents of Smart Cards / Berlin: Springer, 2007. – 337 p.

8. *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.

9. *Trichina E., Korksihko T.* Secure AES Hardware Module for Resource Constrained Devices // Lecture Notes in Computer Science. – Berlin: Springer, 2005. – Vol. 3313. – P. 215 – 229.

10. *Коркішко Л. М.* Базові логічні елементи для комп'ютерних пристроїв захисту інформації// Вісник Національного університету "Львівська політехніка" "Комп'ютерні системи та мережі". – Львів, 2006. – №573 – С. 103 – 113.

11. *Мельник А. О.* Спеціалізовані комп'ютерні системи реального часу. – Львів: Державний університет "Львівська політехніка", 1996. – 54 с.

12. *Черкаський М.* Складність апаратно-програмних комп'ютерних засобів // Сучасні проблеми в комп'ютерних науках. Contemporary Computing in Ukraine CCU'2000. Збірник наукових праць. – Львів, 2000. – С. 58 – 67.

13. *Golic J.D.* Techniques for random masking in hardware // IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 2007. – Vol. 54 (2) – P. 291 – 300.