15. A. Bellezza, "Countermeasures against Side-Channel Attacks for Elliptic Curve Cryptosystems", Cryptology ePrint Archive, 2001/103, 2001.
http://citeseer.ist.psu.edu/bellezza01countermeasures.html

Надійшла 1.09.2004р.

УДК 681.3.06

Лужецький В.А., Сокирук В.В.

ВИКОРИСТАННЯ АРИФМЕТИЧНИХ ОПЕРАЦІЙ ЗА МОДУЛЕМ 2^n ДЛЯ ПОБУДОВИ БЛОКОВИХ СИМЕТРИЧНИХ ШИФРІВ

Вступ

Важливим (а найчастіше і невід'ємним) атрибутом будь-якої системи захисту інформації є блокові симетричні шифри (БСШ). Їхнє широке застосування обумовлене високою швидкістю, криптостійкістю і широким колом розв'язуваних задач [1].

Мережа Фейстеля, на основі якої побудована велика кількість сучасних БСШ, усе ще є досить надійним криптографічним примітивом [2]. Однак важливий і той факт, що велику увагу розробники приділяють пошуку принципово нових схем побудови БСШ. Прикладами тому служать алгоритми Rijndael [3], що переміг у конкурсі AES (Advanced Encryption Standard) і прийнятий як новий стандарт США FIPS-197, і SHACAL-2 [4], рекомендований для всебічного застосування в галузі криптографічного захисту інформації як один з фіналістів міжнародного проекту NESSIE.

Як правило, в алгоритмах БСШ застосовуються різні логічні й арифметичні операції над підблоками блоку даних, що шифрується, з метою перемішування і розсіювання біт відкритого тексту на основі ключової інформації. Такі алгоритми іноді важко представити у вигляді чіткої математичної моделі, що ускладнює аналіз криптостойкости шифру. У даній статті пропонуються алгоритми, які можуть бути базовими при побудові БСШ із простою математичною моделлю і водночас можуть забезпечити високу швидкість шифрування.

Узагальнений підхід до використання арифметичних операцій за модулем

Нехай $Z_m = \{0,1,2,...,m-1\}$ - множина цілих додатних чисел. Операція множення за модулем m чисел $A, B \in Z_m$ описується таким виразом:

$$A \cdot B \equiv C(mod\ m) \qquad (1)$$

Якщо відомі добуток чисел A і B за модулем m і один з множників, наприклад, число B, то для знаходження іншого множника A необхідно виконати операцію ділення за модулем m:

$$A = \left(\frac{C}{B}\right)_{mod\ m} = \left(C \cdot \frac{1}{B}\right)_{mod\ m} = \left(C \cdot \left(\frac{1}{B}\right)_{mod\ m}\right)_{mod\ m} \qquad (2)$$

З теорії чисел відомо [2], що рівняння (2) має єдиний розв'язок тільки якщо існує число $\left(\frac{1}{B}\right)_{mod\ m}$, тобто виконується таке співвідношення:

$$НСД(B,m) = 1 \qquad (3)$$

Для обчислення числа $\left(\frac{1}{B}\right)_{mod\ m}$ може бути використаний розширений алгоритм Евкліда [2], реалізація якого для великих чисел є достатньо складною.

Однак складність операцій множення і ділення за модулем може бути значно зменшена, якщо як модуль використовувати число 2^n, де n – розрядність блоку даних. У разі

використання такого модуля, операція ділення (2) забезпечує єдиний результат, коли K є непарним цілим додатнім числом, тобто виконується умова (3).

Операція множення за модулем 2^n

Відомі алгоритми виконання операції множення за модулем складаються з дій, що реалізують безпосередньо множення, і дій, що визначають остачу від ділення результату (проміжного або остаточного) на модуль [2]. Особливість реалізації операції множення за модулем 2^n полягає в тому, що для отримання остач не потрібні додаткові арифметичні дії. Достатньо лише відкидати цифри розрядів, що мають номери більші за n. Це спрощує реалізацію операції множення за модулем 2^n.

Найвідомішим алгоритмом множення є алгоритм множення „у стовпчик" з послідовними множеннями та розповсюдженням перенесень від молодших до старших розрядів. Він дозволяє представити операцію множення чисел великої розрядності у вигляді послідовності операцій над числами меншої розрядності, якими може ефективно оперувати процесор.

Більшість сучасних комп'ютерів реалізують обчислення над 32- і 64-розрядними числами. Крім того, застосування розширених інструкцій SSE дозволяє так саме ефективно працювати і з числами більшої розрядності.

Нехай n – розрядність операндів і $n = k \cdot m$, де k - кількість m-розрядних слів. Операції множення і додавання в алгоритмі виконуються над числами розрядності m.

Алгоритм множення за модулем 2^n

Вхідні дані	$A = \{A_{k-1}, A_{k-2}, ..., A_0\}$, $B = \{B_{k-1}, B_{k-2}, ..., B_0\}$
Додаткові змінні	$P = \{P_{ст}, P_{мл}\}$
Вихідні дані	$C = \{C_{k-1}, C_{k-2}, ..., C_0\}$
Процедура	*for i = 0 to k-1 {* *for j = 0 to i {* $P = A_j \times B_i$ $C_i = C_i + P_{мл}$ *if i > 1 then* $C_{i-1} = C_{i-1} + P_{ст}$ *}* *}*

Для отримання оцінки обчислювальної складності алгоритму припустимо, що всі операції в алгоритмі виконуються за однаковий час. В загальному випадку знаходження добутку двох n-розрядних чисел зводиться до виконання $\frac{k \cdot (k+1)}{2}$ множень, k^2 додавань і $\frac{k \cdot (k+1) \cdot (2k+1)}{6}$ перенесень m-бітних чисел. Якщо $n = 128$, $k = 4$, то загальна кількість машинних операцій, необхідних для виконання операції: 10 + 16 + 30 = 56. Очевидно, що між часом виконання алгоритму і довжиною числа існує нелінійна залежність.

Таблиця 1

Складність операції множення за модулем 2^n для чисел різної розрядності

n	k	m	Кількість операцій	Кількість машинних операцій на один біт результату
128	4	32	56	0,4375
128	2	64	12	0,09375
256	8	32	304	1,1875
256	4	64	56	0,21875
512	16	32	1888	3,6875
512	8	64	304	0,59375
1024	32	32	12992	12,6875
1024	16	64	1888	1,84375

Існують шляхи суттєво прискорити час виконання алгоритму. Так, в програмних реалізаціях за рахунок використання додаткових змінних для зберігання результатів множення на кожній ітерації можна позбутись великої кількості додавань з перенесенням, на долю яких припадає більша частина операцій в алгоритмі.

Операція ділення за модулем 2^n

Як вже зазначалось, для обчислення числа $\left(\frac{1}{B}\right)_{mod\,2^n}$ може бути використаний розширений алгоритм Евкліда [2], реалізація якого для великих чисел є достатньо складною. Пропонований нижче метод обчислень дозволяє істотно спростити цю процедуру.

Запишемо добуток (1) для двійкового представлення операндів. Будемо нумерувати ваги розрядів від 0 до $n-1$, починаючи з молодших. Так, число A в двійковому вигляді представляється як:

$$A = a_{n-1}\cdot 2^{n-1} + a_{n-2}\cdot 2^{n-2} + \ldots + a_1\cdot 2 + a_0 \qquad (4)$$

Добуток чисел $A = \{a_{n-1}, a_{n-2}, \ldots, a_0\}$ і $B = \{b_{n-1}, b_{n-2}, \ldots, b_0\}$ буде мати такий вигляд:

$$\begin{array}{cccccccc}
 & & & & \times a_{n-1} & \ldots & a_1 & a_0 \\
 & & & & b_{n-1} & \ldots & b_1 & b_0 \\
\hline
 & & & & a_{n-1}b_0 & \ldots & a_1b_0 & a_0b_0 \\
 & & & a_{n-1}b_1 & a_{n-2}b_1 & \ldots & a_0b_1 & \\
 & & \ldots & \ldots & \ldots & \ldots & & \\
 & a_{n-1}b_{n-1} & \ldots & a_1b_{n-1} & a_0b_{n-1} & & & \\
\hline
c_{2n-1} & c_{2n-2} & \ldots & c_n & c_{n-1} & \ldots & c_1 & c_0
\end{array} \qquad (5)$$

Результатом множення цих чисел за модулем 2^n будуть n молодших розрядів результату, значення яких визначаються таким чином:

$$\begin{cases}
c_0 = a_0b_0, \\
c_1 = a_1b_0 + a_0b_1, \\
c_2 = a_2b_0 + a_1b_1 + a_0b_2, \\
\ldots \\
c_{n-1} = a_{n-1}b_0 + a_{n-2}b_1 + \ldots + a_0b_{n-1}.
\end{cases} \qquad (6)$$

Оскільки коефіцієнти $b_0, b_1, ..., b_{n-1}$ і значення $c_0, c_1, ..., c_{n-1}$ відомі, то маємо систему з n рівнянь з n невідомими. Число B непарне, а отже $b_0 = 1$. Тоді систему (6) можна записати в такому вигляді:

$$\begin{cases} c_0 = a_0, \\ c_1 = a_1 + a_0 b_1, \\ c_2 = a_2 + a_1 b_1 + a_0 b_2, \\ \ldots \\ c_{n-1} = a_{n-1} + a_{n-2} b_1 + \ldots + a_0 b_{n-1}. \end{cases} \quad (7)$$

Звідси маємо:

$$\begin{cases} a_0 = c_0, \\ a_1 = c_1 - a_0 b_1, \\ a_2 = c_2 - a_1 b_1 - a_0 b_2, \\ \ldots \\ a_{n-1} = c_{n-1} - a_{n-2} b_1 - \ldots - a_0 b_{n-1}. \end{cases} \quad (8)$$

Таким чином, операція ділення за модулем 2^n зводиться до розв'язання системи з n рівнянь. Необхідною і достатньою умовою існування єдиного розв'язку є $b_0 = 1$.

Алгоритм ділення за модулем 2^n:

Вхідні дані	$B = \{b_{n-1}, b_{n-2}, \ldots, b_0\}$ $C = \{c_{n-1}, c_{n-2}, \ldots, c_0\}$
Додаткові змінні	$S = \{s_{n-1}, s_{n-2}, \ldots, s_0\}$
Вихідні дані	$A = \{a_{n-1}, a_{n-2}, \ldots, a_0\}$
Процедура	*S = 0* *A = 0* *for i = 1 to n {* *if s_i + b_i == c_i then {* *a_i = 1* *S = S + B* *}* *B = B * 2* *}*

Зворотне перетворення приблизно потребує виконання n додавань, n порівнянь, $\frac{k \cdot (k+1)}{2} \cdot m$ додавань з переносом і $\frac{k \cdot (k+1)}{2} \cdot m$ зсувів. Таким чином, при аналогічних параметрах (n = 128, k = 4) обчислювальна складність алгоритму складе 128+128+320+320=896 операцій. Очевидно, цей алгоритм також характеризується нелінійною залежністю часу виконання від розмірності чисел.

Таблиця 2

Складність операції ділення за модулем 2^n для чисел різної розрядності

n	k	m	Кількість операцій	Кількість машинних операцій на один біт результату
128	4	32	896	7
128	2	64	640	5
256	8	32	2816	11
256	4	64	1792	7
512	16	32	9728	19
512	8	64	5632	11
1024	32	32	35840	35
1024	16	64	19456	19

Приведені оцінки можна вважати верхньою границею обчислювальної складності перетворень. Слід зазначити, що описані алгоритми орієнтовані, в першу чергу, на програмну реалізацію. Збільшення розрядності процесора або використання додаткових інструкцій сучасних процесорів для ефективної реалізації арифметичних операцій над великими числами приводить до суттєвого збільшення швидкості обчислень. Як видно з табл. 2, подвоєння розрядності дозволяє збільшити вдвічі розрядність блоку даних із збереженням швидкості обчислень.

Особливості реалізації БСШ на основі описанних операцій

Якщо прийняти, що одним з операндів є n-розрядний секретний ключ, а другим - n-розрядний блок відкритого тексту, описані арифметичні операції за модулем 2^n можна використовувати як базові елементи БСШ. Процедурою зашифрування в даному випадку є операція множення за модулем 2^n. Для розшифрування блоку даних використовується операція ділення за модулем 2^n. Як зазначалось, необхідною і достатньою умовою однозначного відтворення блоку відкритого тексту є вибір як ключа цілого непарного числа необхідної розрядності.

Операції множення і ділення за модулем 2^n можуть бути максимально ефективно реалізовані у програмному вигляді на будь-яких платформах. Такі операції є більш швидкими ніж відповідні їм операції зашифрування та розшифрування багатьох сучасних БСШ. Слід зазначити, що в найпростішій реалізації описані алгоритми не вимагають пам'яті для зберігання додаткової інформації (S-блоків і т.п.), що може спростити апаратну реалізацію.

Простота запропонованих арифметичних операцій також є важливим плюсом і може бути ефективно використана для оцінки криптографічної стійкості БСШ, побудованих на їх основі. Задача знаходження відкритого тексту без знання ключової інформації зводиться до задачі розкладання великого складного числа на складні множники. Розкладання на множники добутку двох складених чисел є задачею більш складної, ніж задача розкладання на множники добутку двох простих чисел, для розв'язання якої і досі не знайдено ефективних алгоритмів. Зважаючи на те, що криптоаналітику відомий лише модуль добутку, варіантів розкладання може бути безліч, тому однозначне відновлення множників можливо тільки при знанні одного з них.

Висновки

Арифметичні операції множення та ділення за модулем 2^n можуть бути використані як базові елементи БСШ. Вони дозволяють однаково ефективно реалізовувати БСШ із різною (у тому числі і змінною) довжиною блоку. Від розміру блоку залежить тільки швидкість виконання перетворень, сам алгоритм і його деталі залишаються незмінними.

Проста математична модель перетворення істотно спрощує аналіз криптостійкості і гарантує відсутність "лазівок". Алгоритм може бути ефективно реалізований на різних

платформах, а обмежений набір застосовуваних операцій дозволяє сподіватися на ефективну апаратну реалізацію.

Список літератури

1. *Вильям Столлингс*. Криптография и защита сетей: принципы и практика, 2-е изд. – М.: Издательский дом "Вильямс", 2001. – 672с.

2. *Петров А.А.* Компьютерная безопасность. Криптографические методы защиты. – М.: ДМК, 2000. – 448с.

3. *Joan Daemen, Vincent Rijmen*. The Rijndael Block Cipher. AES Proposal: Rijndael, Document version 2, 3.09.99.

4. *Гулак Г.М., Горбенко И.Д., Михайленко М.С., Гитис Ю.Є.* Блочний симетричний криптоалгоритм SHACAL-2 // "Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні", випуск – 7, 2003 р, 224 с.

Надійшла 13.10.2004р.

УДК 681.3.06(075)

С.Р.Коженевский

ОСОБЕННОСТИ ВОССТАНОВЛЕНИЯ ИНФОРМАЦИИ, ХРАНИМОЙ НА ЖЕСТКИХ ДИСКАХ

В компьютерных системах наиболее ценная и важная информация хранится в накопителях на жестких магнитных дисках (НЖМД). Это обуславливается их технико-экономическими показателями, такими как: энергонезависимость, простота в использовании, большие объемы данных при низкой стоимости хранения единицы информации, высокая скорость записи и считывания. Поэтому потеря данных, хранящихся на НЖМД, всегда очень болезненна для пользователей.

Широкому применению накопителей на жестких дисках способствует ряд его положительных эксплуатационных качеств: надежность, быстрота доступа и дешевизна в расчете на единицу хранения информации. Кроме того, один из самых важных показателей - энергонезависимость делает НЖМД практически незаменимым для оперативного и долговременного хранения больших массивов информации.

В то же время, размещение и хранение информации в устройствах долговременной энергонезависимой памяти создает предпосылки как для утраты важной информации, так и для несанкционированного доступа к ней.

Потеря информации на НЖМД может происходить вследствие износа рабочих поверхностей пластин жесткого диска, что приводит к нарушению служебных областей диска и области данных, либо вследствие его поломки. Кроме того, потеря данных может произойти и на полностью исправном НЖМД вследствие некомпетентных действий пользователей, системных сбоев, воздействий вирусов. Во всех этих случаях результат один - невозможность считывания информации штатными средствами.

Ценность утерянной информации изменяется в широких пределах: от малоценной до бесценной. В ряде случаев ценность утерянной информации настолько велика, что заказчик готов платить любые деньги за ее восстановление.

В последнее время значительно увеличился объем жестких дисков. В основном увеличение объема достигнуто за счет увеличения плотности записи. Увеличение плотности записи привело к необходимости применения специальных мер, направленных на увеличение надежности жестких дисков. Несмотря на принимаемые производителями жестких дисков меры по обеспечению надежности, жесткий диск остается самым ненадежным элементом компьютера. Ежегодно в сервисный центр «ЕПОС» поступает для ремонта более полутора - двух тысяч жестких дисков. Примерно треть из них имели