Список литературы:

1. *Азарова О.В.* Анализ рисков системе конкурентной разведки. / Сборник научных трудов НАУ «Защита информации» - К.: Издательство НАУ 2004. с.34-45.

2. *Андреев В.И., Козлов В.С., Хорошко В.А.* Количественная оценка защищенности технических объектов с учетом их функционирования. / Захист інформації №2, 2004. с.47-51.

3. *Кальманс А.К.* О выборе оптимальной вершины в графе. / В кн. «Исследования по дискретной математике». – М.: Наука, 1989. с. 151-158.

4. *Замбицкий Д.К., Солтан П.С.* Об одной экстремальной задаче на дереве. / В кн. «Математические методы решения экономических задач». Вып. 11. – М.: Наука, 1989. с. 102-107.

5. *Замбицкий Д.К.* Относительно одной экстремальной задачи на графе. / В кн. «Прикладная математика и программирование». – Кишинев: РИО АК МССР, 1988. с. 18-27.

6. *Духовный М.А.* Об одной оптимальной задаче теории графов. / В кн. «Математические заметки». Вып. 3. – М.: Наука, 1981, 10. с. 355-359.

7. *Хорошко В.О., Кудінов В.А.* методичний підхід формалізації задачі оцінювання ефективності системи захисту інформаційної системи ОВС України. / Захист інформації, №4, 2004. с. 11-18.

8. *Сибирский В.К.* Параметрическая задача Штейнера на графах. / Известия АН МССР. Сер. физ-техн. и математ. наук, №3, 1986. с. 22-25.

9. *Гольштейн Е.Г., Юдин Д.Б* Задачи линейного программирования транспортного типа. – М.: Наука, 1989. 254с.

Поступила 20.10.2004г.

УДК 681.188; 004.056.5; 004.421.2:517.518.26

Васильцов І.В., Васильків Л.О.

СТІЙКІСТЬ СУЧАСНИХ АЛГОРИТМІВ МОДУЛЯРНОГО ЕКСПОНЕНЦІЮВАННЯ ДО ЧАСОВОГО АНАЛІЗУ

Вступ

На сучасному етапі розвитку криптографії існує багато різноманітних методів та засобів захисту інформації. Поширеними та популярними стали методи захисту, які базуються на асиметричній криптографії, оскільки її застосування дозволило вирішити задачу розподілу ключів та електронного цифрового підпису [1-5].

Асиметричні криптосистеми на сьогоднішній день досліджені менше, ніж симетричні, оскільки концепція їх побудови була запропонована відносно недавно.

Вагомою причиною, яка не дозволяє використовувати асиметричні алгоритми шифрування для захисту інформації в системах передачі даних – це низька швидкодія виконання основних математичних процедур, які використовуються для шифрування та дешифрування. Цей факт особливо характерний для реалізацій алгоритмів на пристроях із невеликими обчислювальними можливостями.

Одним із шляхів вирішення цієї проблеми є зменшення розмірності параметрів системи, але це може привести до зменшення стійкості системи. Іншою можливістю є застосування алгоритмів, призначених для ефективного здійснення основних математичних обчислень, які використовуються в асиметричних криптосистемах [2,6,7]. При цьому

ефективність, залежно від ситуації, може трактуватися не тільки як висока швидкість роботи, але і як мінімальний об'єм пам'яті, мінімальний програмний код та їх сукупність.

Як відомо, здійснення операцій шифрування та дешифрування в асиметричних криптосистемах вимагає обчислення відповідної до алгоритму важкооборотної функції. Тому найбільшого часу вимагають операції множення за модулем та піднесення до степеня за модулем. Безпосереднє здійснення вказаних операцій вимагає дуже багато часу та великого об'єму пам'яті, що негативно впливає на загальний час обробки інформації. Сучасні алгоритми модулярного експоненціювання базовані лише на операціях модулярного додавання, віднімання, множення, порозрядного зсуву чи звичайного множення або додавання. Тому вони суттєво прискорюють процес шифрування та дешифрування і криптосистеми з їх використанням можна реалізовувати на пристроях із невеликими обчислювальними можливостями.

В умовах розвитку сучасних інформаційних технологій задача захисту інформаційних ресурсів володіє певними особливостями. Природно, що для її вирішення необхідно використовувати апаратну реалізацію відомих алгоритмів криптографічного захисту інформації. Проте вимоги, які ставляться до них на сучасному етапі, обумовили появу принципово нових видів криптоаналізу, які умовно можна назвати "Атаки спеціальних впливів", або ж "Атаки на основі нестандартних (побічних) каналів витоку інформації" (англ. мовою *side-channel attacks, covert-channel attacks*) [2, 3, 5, 8-13]. Тому розробка підходів, методів, алгоритмів та засобів проектування криптографічних пристроїв захисту інформації, що є стійкими до такого виду атак є важливою та актуальною задачею.

Час виконання криптографічних операцій залежить не лише від ефективності реалізації конкретного алгоритму, але й також (інколи суттєво) від вхідних даних. Особливо така кореляція сильно проявляється для алгоритмів модулярного експоненціювання асиметричних криптосистем та алгоритмів додавання (множення) точок на еліптичній кривій [12, 14]. Як правило, ці криптографічні операції є обчислювально складними і для підвищення продуктивності виконання процедури шифрування повідомлення, чи формування цифрового підпису використовують спеціальні алгоритми, які базуються на оцінці інформації про кількість бітів у ключі шифрування [2, 6, 7]. Така оцінка в алгоритмах дозволяє пришвидшити виконання криптографічних операцій за рахунок обходу виконання деяких операцій алгоритму при нульових бітах ключа, що дає змогу значно підвищити продуктивність засобів захисту інформації. Звідси очевидно, що завжди можна виявити певну кореляцію між кількістю одиничних бітів ключа та часом виконання такого алгоритму. Саме така інформація дозволяє зловмиснику висунути гіпотезу щодо кількості одиничних та нульових бітів у секретному ключі, кількісним еквівалентом якої може бути вага Хемінга, а вже на основі такої оцінки здійснити атаку повного перебору в певному під-діапазоні ключового простору, що потребує значно менших обчислювальних ресурсів [3, 5, 12, 15]. Таким чином оцінка кореляції часу виконання криптографічних операцій до ваги Хемінга ключової інформації дозволяє зловмиснику зменшити обчислювальну складність атаки на систему захисту інформаційних ресурсів. Атаки такого виду називають "*часовим аналізом*".

Ефективність та стійкість криптосистеми до атак є конфліктними цілями. З одного боку, постійно підвищуються вимоги до безпеки, оскільки алгоритм повинен мати запас стійкості не тільки до відомих криптоатак, але і до нових методів криптоаналізу. З іншого боку, підвищуються вимоги до продуктивності засобів шифрування. Повне вирішення протиріч між стійкістю і ефективністю отримати неможливо, проте якщо розглядати стійкість як складову ефективності алгоритму та узгодити різні показники, гостроту вирішення цього питання можна знизити.

Метою роботи є дослідження стійкості відомих алгоритмів модулярного експоненціювання до часової аналізу, що дозволить виявити важливі особливості їх реалізації, а також розробити рекомендації стосовно побудови сучасних систем захисту інформації, що базуються на асиметричній криптографії.

1. Методи піднесення числа до степеня за модулем та алгоритми їх реалізації

Багато криптосистем з відкритим ключем використовують функцію дискретного піднесення до степеня

$$f(n) = x^n (mod\, m), \qquad (1)$$

де n – ціле число $(1 \le n \le m-1)$, m – велике просте число, x – ціле число $(1 \le x \le m)$.

Оскільки для забезпечення стійкості двоключових систем необхідно використовувати досить великі значення x та p, то виникла потреба у використанні спеціальних методів для спрощення і прискорення процесу обчислення цієї функції. На даний час найчастіше вживаними є бінарний, β-арний методи, метод ковзаючого вікна, а також методи з фіксованим показником, з фіксованою основою та з використанням особливостей модулів [2].

Бінарний метод використовує двійкове (бінарне) зображення числа $n = (n_{k-1} \dots n_0)_2$. Цей метод виконується у двох напрямах. При зчитуванні *"зліва направо"* x^n записується як:

$$x^n = x^{(n_{k-1} \dots n_0)_2} = \left(\dots \left(\left(x^{n_{k-1}} \right)^2 x^{n_{k-2}} \right)^2 \dots \right) x^{n_1} \Bigg)^2 x^{n_0}. \qquad (2)$$

У бінарному методі на основі зчитування *"справа наліво"* використовується запис:

$$x^n = x^{(n_{k-1} \dots n_0)_2} = \left(x^{2^0} \right)^{n_0} \left(x^{2^1} \right)^{n_1} \dots \left(x^{2^{k-1}} \right)^{n_{k-1}} = \prod_{\{i | n_i = 1\}} x^{2^i}. \qquad (3)$$

Алгоритми реалізації цих методів потребують $\lceil log\, n \rceil$ піднесень до квадрата та $H(n)$ (вага Хемінга, яка дорівнює кількості одиниць в двійковому зображенні числа n) і, отже, $2\lceil log\, n \rceil$ множень у найгіршому випадку та $\frac{3\lceil log\, n \rceil}{2}$ множень у середньому [2].

β-арний метод ґрунтується на зображенні показника степеня за основою β, тобто $n = (n_{k-1} \dots n_0)_\beta$.

Цей метод також виконується в двох напрямках.

При зчитуванні *"зліва направо"*

$$x^n = x^{(n_{k-1} \dots n_0)_\beta} = \left(\dots \left(\left(x^{n_{k-1}} \right)^\beta x^{n_{k-2}} \right)^\beta \dots \right) x^{n_1} \Bigg)^\beta x^{n_0}. \qquad (4)$$

Якщо β є степенем двійки, тобто $\beta = 2^w$ для будь-якого цілого додатного w, то піднесення y^β потребує w піднесень до квадрата. Тоді потрібне лише двійкове зображення числа n, w бітів якого обробляємо за одну ітерацію, рухаючись зліва направо. При $w = 1$ отримуємо бінарний метод "зліва направо".

Для $\beta = 2^w$ потрібно виконати в алгоритмі реалізації даного методу $2^w - 1 + \frac{\lceil log\, n \rceil}{w}$ множень та $\frac{\lceil log\, n \rceil}{w}$ піднесень до степеня. Отже, кількість множень щонайбільше $2^w - 1 + \frac{2}{w}\lceil log\, n \rceil$ [2].

При зчитуванні *"справа наліво"*

$$x^n = x^{(n_{k-1}\ldots n_0)_\beta} = \left(x^{\beta^0}\right)^{n_0}\left(x^{\beta^1}\right)^{n_1}\ldots\left(x^{\beta^{k-1}}\right)^{n_{k-1}} = \prod_{w=1}^{\beta-1}\left(\prod_{\{i|n_i=w\}} x^{\beta}\right)^{w}. \quad (5)$$

Для обчислення цього добутку необхідно виконати $2\beta-2$ множень в загальному. Тому в даному алгоритмі "справа наліво" в загальному випадку при $\beta = 2^w$ виконується $2\beta-2+\frac{2}{w}\lceil log\, n\rceil$.

β-арний метод при $\beta = 2^w$, де $w \geq 1$, часто називають методом вікна для піднесення до степеня.

Метод ковзаючого вікна ґрунтується на довільному розбитті на блоки (вікна) бінарного зображення показника степеня, тобто $n = [w_{i-1},\ldots,w_0]_2$. У даному методі вікна не повинні мати однаковий розмір.

Турбер [2] розглядав два типи вікон:

1) нульові вікна, які утворюються лише бітом 0;

2) непарні вікна довжини щонайбільше w, які починаються та закінчуються бітом 1.

При зчитуванні "*зліва направо*" бінарного зображення числа n

$$x^n = \left(\left(\ldots\left(\left(x^{(w_{i-1})_2}\right)^{2^{|w_i-2|}}\cdot x^{(w_{i-2})_2}\right)^{2^{|w_i-3|}}\ldots\right)^{2^{|w_1|}}\cdot x^{(w_1)_2}\right)^{2^{|w_0|}}\cdot x^{(w_0)_2} \quad (6)$$

У алгоритмі реалізації методу ковзаючого вікна "зліва направо" виконується $2^w-1+|w_i|$ множень, де $|w_i|$ – довжина непарного w_i вікна, та $\lceil log\, n\rceil$ піднесень до квадрату.

При зчитуванні "*справа наліво*"

$$x^n = \prod_{i=0}^{l-1} x^{(w_i)_2\cdot 2^{l_i}} = \prod_{w\in\{1,3,\ldots,2^w-1\}}\left(\prod_{\{i|(w_i)_2=w\}} x^{2^{l_i}}\right)^{w}, \quad (7)$$

де $l_i = \sum_{j=0}^{i-1}|w_j|$, для будь-якого $1 \leq i \leq l-1$, $l_0 = 0$.

У загальному випадку в алгоритмі "справа наліво" необхідно виконати $2^w-2+|w_i|$ множень ($|w_i|$ – довжина непарного w_i вікна) та $2^{w-1}-1+\lceil log\, n\rceil$ піднесень до квадрату.

2. Затрати часу на виконання кожного методу

На виконання кожного з вище описаних алгоритмів необхідно затратити певний час. Виконання однієї операції алгоритму залежить від швидкодії процесору, тому можна сказати, що в загальному кожен окремий крок алгоритму виконується за певний час. Основні операції та затрати часу на виконання кожної з них можна подати у вигляді таблиці:

Таблиця 1. Затрати часу на виконання основних операцій алгоритмів експоненціювання

Операція	Час, в тактах	Зміст операції
$a = b$	c	*Просте присвоєння*
$z = x \bmod m$	b	*Присвоєння за модулем*
$\mathrm{FIND}(max\{n_i,..,n_j\} \mid i-j+1 \le w, n_j = 1)$	q	*Знаходження найдовшої послідовності бітів такої, що* $i-j+1 \le w$ та $n_j = 1$
$n = (n_{k-1} \ldots n_0)_2$	t	*Зображення числа в двійковій системі числення*
$y = x \cdot x \bmod m$	r	*Піднесення до квадрату за модулем*
$z = x \cdot y \bmod m$	s	*Множення за модулем*
$z = y^{\beta} \bmod m$	d	*Піднесення до степеня за модулем*

В загальному можна прийняти, що співвідношення між величинами значень цих часів є таким:

$$c \le b \le q \le t \le r \le s \le d . \tag{8}$$

Виходячи з даних у таблиці можна побудувати математичну модель обчислення часу, затраченого на виконання кожного з алгоритмів виконання методів, описаних вище.

На виконання бінарного методу затрачається час:

при зчитуванні "зліва направо":

$$T1(n) = t + c + \sum_{i=k-1}^{0} r_i + \sum_{\{i \mid n_{i=1}\}} s_i = t + c + \lceil log\, n \rceil \cdot r + H(n) \cdot s \tag{9}$$

– при зчитуванні "справа наліво":

$$T2(n) = t + c + b + \sum_{\{i \mid n_{i=1}\}} s_i + \sum_{i=0}^{k-1} r_i = t + c + b + H(n) \cdot s + \lceil log\, n \rceil \cdot r \tag{10}$$

На виконання β-арного методу затрачається час:

– при зчитуванні "зліва направо":

$$\begin{aligned} T3(n,w) &= t + c + \sum_{i=1}^{\beta-1} s_i + c + \sum_{i=k-1}^{0} (d_i + s_i) = \\ &= t + 2c + \left(\frac{\lceil log\, n \rceil}{w} + 2^w - 1 \right) \cdot s + \frac{\lceil log\, n \rceil}{w} \cdot d \end{aligned} \tag{11}$$

– при зчитуванні "справа наліво":

$$\begin{aligned} T4(n,w) &= t + b + \sum_{w=1}^{\beta-1} c_w + \sum_{0}^{k-1} \left(d_{\{i \mid n_i = 0\}} + s_{\{i \mid n_i = 1\}} + d_{\{i \mid n_i = 1\}} \right) + 2c + \sum_{w=\beta-1}^{1} 2 s_w = \\ &= t + \left(2^w + 1 \right) c + b + \frac{\lceil log\, n \rceil}{w} \cdot d + \left(\frac{\lceil log\, n \rceil}{w} - W_0(n) + 2^{w+1} - 2 \right) \cdot s \end{aligned} \tag{12}$$

де $W_0(n)$ – кількість нульових бітів у зображенні числа n за основою β.

Очевидно, що в бінарному зображенні числа n є $\lceil \log n \rceil - H(n)$ нульових бітів. Для переведення числа в β-арну систему числення бінарне зображення n розбивають на вікна довжиною w. Звідси випливає, що верхня оцінка:

$$W_0{}^{max}(n) = \left\lfloor \frac{\lceil \log n \rceil - H(n)}{w} \right\rfloor. \tag{13}$$

З іншого боку, нижня оцінка легко може бути визначена як

$$W_0{}^{min}(n) = \left\lfloor \frac{(\lceil \log n \rceil - H(n)) \cdot w}{(w-1) \cdot \lceil \log n \rceil} \right\rfloor, \tag{14}$$

На виконання методу ковзаючого вікна затрачається час:

– при зчитуванні "зліва направо":

$$\begin{aligned} T5(n,|w_i|) &= b + s + \sum_{j=1}^{2^{|w_i|}-1} s_j + t + 2c + \\ &+ \sum_{i=0}^{k-1} \left((r+c)_{\{i|n_i=0\}} + (q+s+c+r)_{\{i|n_i\neq 0\}} \right) = \\ &= b + s + \left(2^{|w_i|} - 1\right)s + t + 2c + \\ &+ (k - H(n))(r+c) + p(q+s+c) + r(|w_o| + \ldots + |w_i|) = \\ &= t + b + 2c + kr + 2^{|w_i|}s + p(q+s+c) + (k - H(n))c = \\ &= t + b + (2 + p + \lceil \log n \rceil - H(n))c + \lceil \log n \rceil r + \left(2^{|w_i|} + p\right)s + pq \end{aligned} \tag{15}$$

– при зчитуванні "справа наліво":

$$\begin{aligned} T6(n,|w_i|) &= t + b + \sum_{\{j=1,3,\ldots,2^{|w_i|}-1\}} c_j + c + \\ &+ \sum_{i=k-1}^{0} \left((r+c)_{\{i|n_i=0\}} + (q+s+c+d)_{\{i|n_i\neq 0\}} \right) + \\ &+ \sum_{\{v=2^{|w_i|}-1,\ldots,5,3\}} (2s_v) + c = \\ &= t + b + \left(2^{2|w_i|-2} + 1\right)c + (k - H(n))(r+c) + \\ &+ p(q+s+c+d) + 2^{2|w_i|-1}s + c = \\ &= t + b + \left(2^{2|w_i|-2} + 2 + \lceil \log n \rceil - H(n) + p\right)c + \\ &+ (\lceil \log n \rceil - H(n))r + \left(2^{2|w_i|-1} + p\right)s + pq + pd \end{aligned}, \tag{16}$$

де p – кількість вікон, а $(|w_0| + \ldots + |w_i|)$ – сума всіх непарних вікон, яка рівна вазі Хемінга, оскільки ці вікна складаються лише з одиничних бітів.

Очевидно, що $p_{max} = \left\lceil \frac{\log n}{2} \right\rceil$, а $p_{min} = \left\lceil \frac{H(n)}{w_i} \right\rceil$. Отже, в загальному для дослідження часу виконання цього алгоритму можна розглядати середнє значення

$$p = \frac{\left\lceil \frac{H(n)}{w_i} \right\rceil + \left\lceil \frac{\log n}{2} \right\rceil}{2}. \tag{17}$$

3. Оцінка продуктивних характеристик алгоритмів модулярного експоненціювання

Очевидно, що виникає необхідність визначення найпродуктивнішого алгоритму з усіх відомих алгоритмів модулярного експоненціювання для підвищення продуктивності асиметричних криптосистем, де вони використовуються.

Шлях розв'язання цієї задачі розглянемо на прикладі описаних вище бінарного, β-арного методів та методу ковзаючого вікна.

Як зазначалося вище, загальний час виконання алгоритму бінарного методу залежить лише від довжини двійкового зображення числа n. Час виконання алгоритму β-арного методу залежить не тільки від довжини бінарного зображення числа n, а й від значення β (тобто від числа w). Час, який займає виконання алгоритму методу ковзаючого вікна, залежить від довжини двійкового зображення числа n та ширини непарного вікна. Враховуючи це, можна дослідити залежність часу виконання алгоритму $T_i(n, w, w_i)$ від довжини двійкового зображення числа n.

На рисунку 1 зображено цю залежність при усереднених значеннях ваги Хемінга ($H(n)$) та кількості нулів у β-арному зображенні числа n ($W_0(n)$), а також при різних значеннях w, ширини непарного вікна та значеннях $c=1$, $b=1.5$, $q=1.6$, $t=1.6$, $r=15$, $s=16$, $d=19$.

Аналіз рисунку 1 показує, що час виконання алгоритмів модулярного експоненціювання має лінійний характер. Крім того, найпродуктивнішими є алгоритми β-арного методу "зліва направо" та "справа наліво", а найбільше часу займає виконання алгоритму бінарного методу.

На рисунках 2 та 3 зображено залежність швидкодії алгоритмів β-арного методу "зліва направо" та "справа наліво", відповідно, від значення степеня основи w в залежності від різної довжини ключа та при усередненому значенні ваги Хемінга. За даними цих графіків можна визначити оптимальну основу, при якій здійснюється найменша затримка роботи алгоритму, а отже, забезпечується його максимальна продуктивність при заданих значеннях експоненти n.

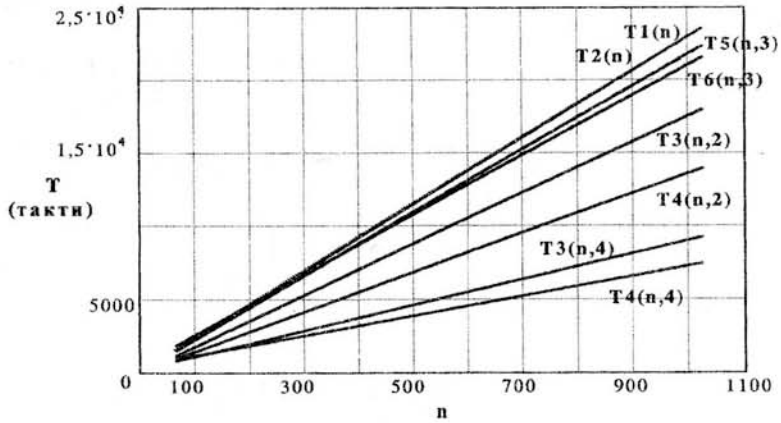

Рис.1 – Оцінка продуктивних характеристик досліджуваних алгоритмів

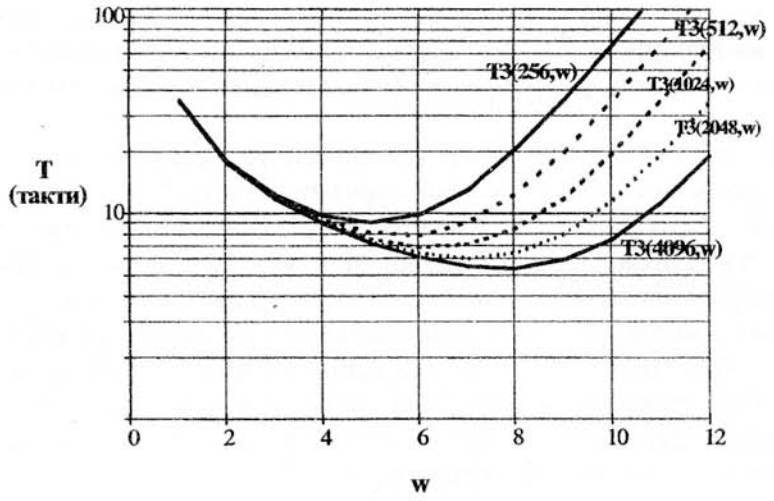

Рис.2 – Залежність швидкодії алгоритму β-арного методу "зліва направо" від значення степеня основи w

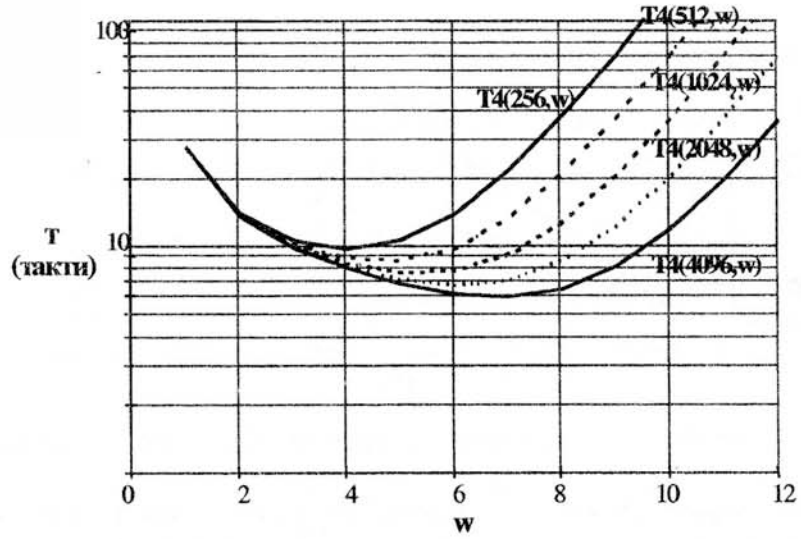

Рис.3 – Залежність швидкодії алгоритму β-арного методу "справа наліво" від значення степеня основи w

Для алгоритмів β-арного методу найкращими будуть значення w, подані в таблиці 2.

Таблиця 2 – Оптимальні значення степеня основи β-арного методу при різній довжині ключа n.

n	w	
	β-арний метод "зліва направо"	β-арний метод "справа наліво"
4096	8	7
2048	7	6
1024	6	5
512	6	5
256	5	4

4. Вага Хемінга як критерій оцінки чутливості до атак спеціального виду

Як було вказано вище, виявлення певної кореляції між кількістю одиничних бітів ключа та часом виконання відповідного алгоритму дозволяє зловмиснику висунути гіпотезу щодо цієї кількості одиничних (нульових) бітів, кількісним еквівалентом якої може бути вага Хемінга.

Тому для дослідження стійкості алгоритмів, описаних вище, необхідно встановити залежність часу виконання відповідного алгоритму від ваги Хемінга.

На рисунку 4 зображено залежність часу виконання алгоритмів бінарного методу "зліва направо" та "справа наліво", відповідно, від ваги Хемінга при довжині $n = 1024$. Аналіз цього графіку показує, що продуктивність даних алгоритмів суттєво залежить від ваги Хемінга, тобто можна визначити мінімальну та максимальну швидкодію, математичне сподівання і т.п. Крім того, очевидно, що стійкість цих методів до часового аналізу буде мінімальною. Тобто, зловмисник, вимірявши час виконання алгоритму, зможе легко оцінити кількість одиниць у двійковому зображенні числа n, а отже і визначити таємний ключ шляхом перебору у звуженому ключовому просторі.

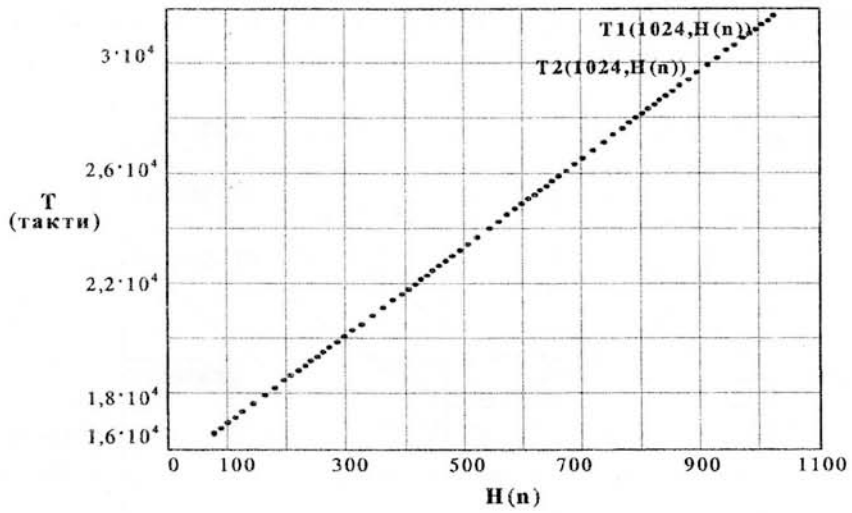

Рис.4 – Залежність часу виконання алгоритму бінарного методу від ваги Хемінга

Аналіз графіку залежності швидкодії алгоритму β-арного методу "зліва направо" від ваги Хемінга (рисунок 5) показує, що, на відміну від бінарного методу (рисунок 4), час виконання цього алгоритму залежить лише від значення β. Тобто цей алгоритм є абсолютно стійкий до часової атаки.

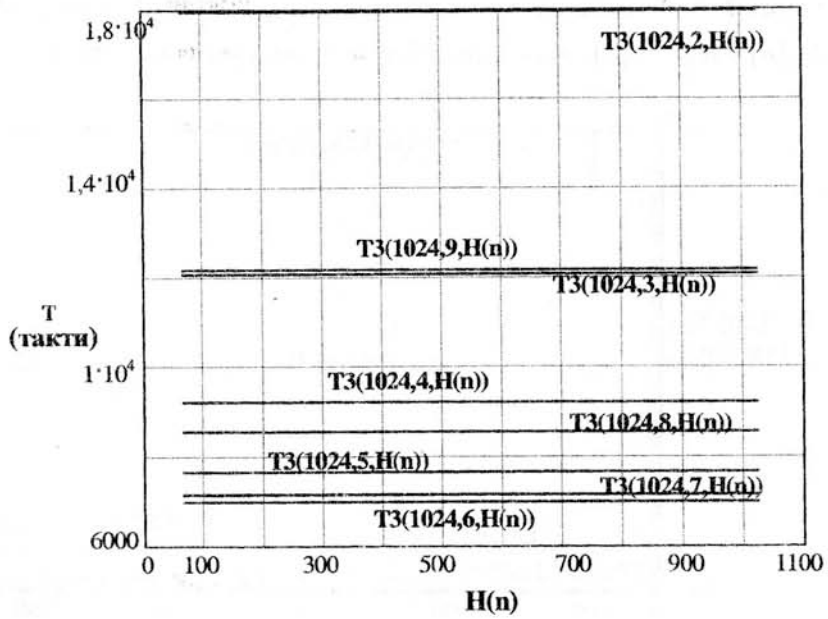

Рис.5 – Залежність часу виконання алгоритму β-арного методу "зліва направо" від ваги Хемінга

Дослідження залежності часу виконання алгоритму β-арного методу "справа наліво" від ваги Хемінга (рисунок 6) показує, що на відміну від попереднього (рисунок 5), він залежить від кількості одиниць у двійковому зображенні числа n. Тобто при різних значеннях його параметрів отримуються різні характеристики швидкодії та стійкості до часового аналізу. Проте в окремих випадках можна знайти таке значення β, при якому можливе отримання практичної стійкості, близьке до абсолютної, наприклад, при $\beta = 9$.

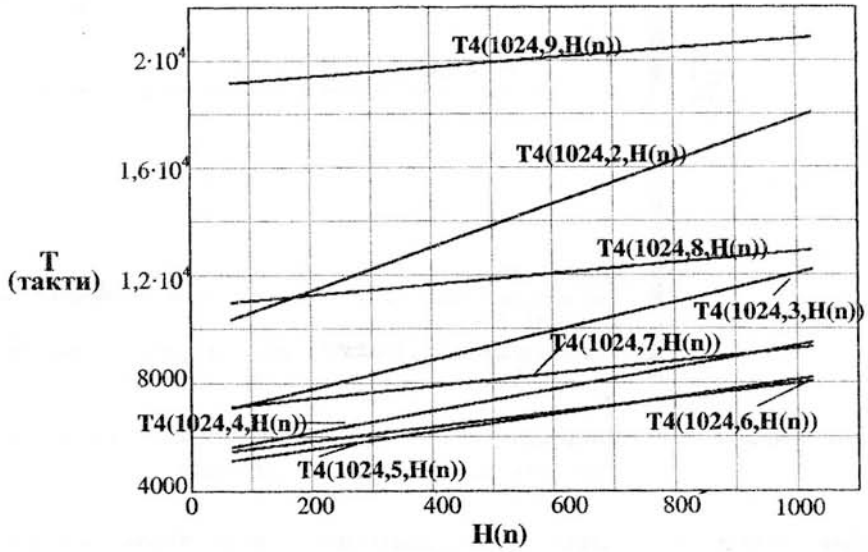

Рис.6 – Залежність швидкодії алгоритму β-арного методу "справа наліво" від ваги Хемінга

З аналітичного представлення формул (15) та (16) випливає, що існує певна обернена залежність часу виконання алгоритмів методу ковзаючого вікна при зчитуванні "зліва направо" та "справа наліво", відповідно, від ваги Хемінга. Проте, оскільки ця залежність є відносно невеликою, то можна вважати, що для певного класу прикладних задач можна успішно використовувати вказані алгоритми, оскільки їхня стійкість до часового аналізу є вищою в порівнянні з іншими методами.

На рисунках 7 та 8 зображено залежність часу виконання цих алгоритмів від ваги Хемінга, при $W_0(n) = W_0{}^{max}(n)$, що є найсприятливішою умовою для криптоаналізу.

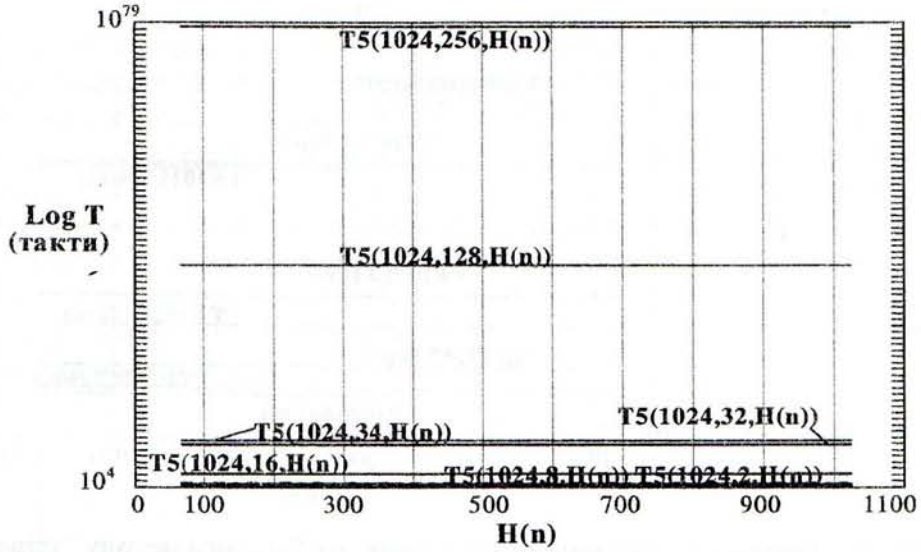

Рис.7 – Залежність швидкодії алгоритму методу ковзаючого вікна при зчитуванні "зліва направо" від ваги Хемінга

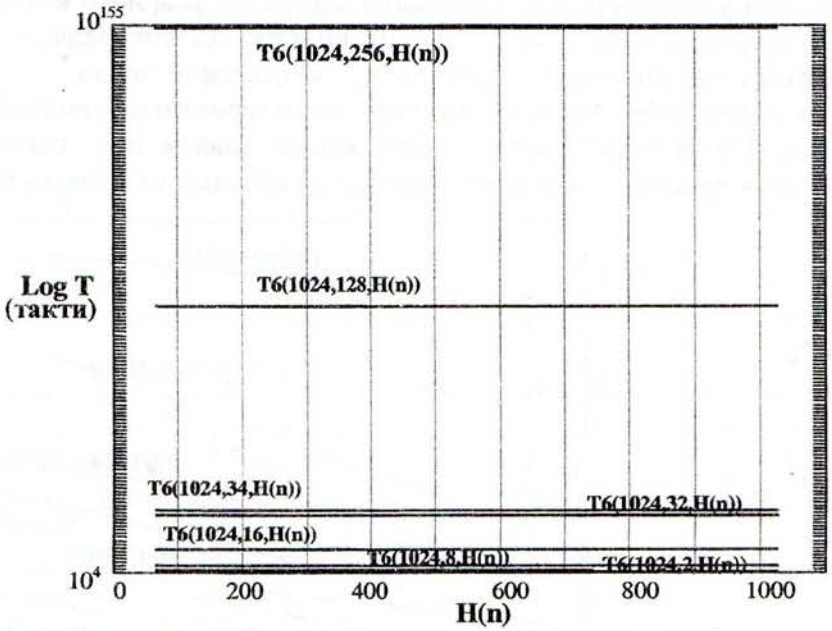

Рис.8 – Залежність швидкодії алгоритму методу ковзаючого вікна при зчитуванні "справа наліво" від ваги Хемінга

5. Оцінка стійкості досліджуваних алгоритмів модулярного експоненціювання до часового аналізу

Для оцінки стійкості розглянутих алгоритмів необхідно продиференціювати математичні моделі часу виконання кожного алгоритму за основою $H(n)$. В результаті отримаємо:

$$\frac{dT1}{dH(n)} = \frac{d\left(t + c + \lceil \log n \rceil \cdot r + H(n) \cdot s\right)}{dH(n)} = s \qquad (18)$$

$$\frac{dT2}{dH(n)} = \frac{d(t + c + b + \lceil log\, n \rceil \cdot r + H(n) \cdot s)}{dH(n)} = s \tag{19}$$

$$\frac{dT3}{dH(n)} = \frac{d\left(t + 2c + \frac{\lceil log\, n \rceil}{w} \cdot d + \left(2^w - 1 + \frac{\lceil log\, n \rceil}{w}\right) \cdot s\right)}{dH(n)} = 0 \tag{20}$$

$$\frac{dT4}{dH(n)} = \frac{d\left(t + \left(2^w + 1\right) \cdot c + b + \frac{\lceil log\, n \rceil}{w} \cdot d + \left(2^{w+1} - 2 - W_0(n) + \frac{\lceil log\, n \rceil}{w}\right) \cdot s\right)}{dH(n)} =$$

(21)

$$= \frac{-d(W_0(n) \cdot s)}{dH(n)} = \frac{-d\left(\left\lfloor \frac{\lceil log\, n \rceil - H(n)}{w} \right\rfloor \cdot s\right)}{dH(n)} = \frac{s}{w}$$

$$\frac{dT5}{dH(n)} = \frac{d\left(t + (n + 2 + p - H(n)) \cdot c + b + \lceil \log n \rceil \cdot r + \left|2^{w_i} + p\right| \cdot s + p \cdot q\right)}{dH(n)} =$$

$$= \frac{d((p - H(n)) \cdot c)}{dH(n)} + \frac{d(p \cdot s)}{dH(n)} + \frac{d(p \cdot q)}{dH(n)} =$$

$$= \frac{d\left(\frac{\left|\frac{H(n)}{w_i}\right| + \lceil \log n \rceil}{2} - H(n) \cdot c\right)}{dH(n)} + \frac{d\left(\left\lceil \frac{\left|\frac{H(n)}{w_i}\right| + \lceil \log n \rceil}{2} \right\rceil \cdot s\right)}{dH(n)} +$$

$$+ \frac{d\left(\frac{\left|\frac{H(n)}{w_i}\right| + \lceil \log n \rceil}{2} \cdot q\right)}{dH(n)} =$$

$$= \left(\left|\frac{1}{2w_i}\right| - 1\right) \cdot c + \left|\frac{1}{2w_i}\right| \cdot s + \left|\frac{1}{2w_i}\right| \cdot q = \tag{21}$$

$$= \left|\frac{1}{2w_i}\right| \cdot (c + s + q) - c$$

$$\frac{dT6}{dH(n)}=\frac{d\left(t+\left(\lceil logn\rceil+2+p-H(n)+2^{2w_i-2}\right)\cdot c+b+\left(\lceil logn\rceil-H(n)\right)\cdot r+\left\lceil 2^{2w_i-1}+p\right\rceil\cdot s+p\cdot q+p\cdot d\right)}{dH(n)}=$$

$$=\frac{d\left((p-H(n))\cdot c\right)}{dH(n)}+\frac{d\left(\left(\lceil logn\rceil-H(n)\right)\cdot r\right)}{dH(n)}+\frac{d(p\cdot s)}{dH(n)}+\frac{d(p\cdot q)}{dH(n)}+\frac{d(p\cdot d)}{dH(n)}=$$

$$=\frac{d\left(\left\lceil\frac{\left\lceil\frac{H(n)}{w_i}\right\rceil+\lceil logn\rceil}{2}\right\rceil-H(n)\cdot c\right)}{dH(n)}+\frac{d\left(\left\lceil\frac{\left\lceil\frac{H(n)}{w_i}\right\rceil+\lceil logn\rceil}{2}\right\rceil\cdot s\right)}{dH(n)}+$$

$$+\frac{d(-H(n)\cdot r)}{dH(n)}+\frac{d\left(\left\lceil\frac{\left\lceil\frac{H(n)}{w_i}\right\rceil+\lceil logn\rceil}{2}\right\rceil\cdot(q+d)\right)}{dH(n)}=$$

$$=\left(\left|\frac{1}{2w_i}\right|-1\right)\cdot c-r+\left|\frac{1}{2w_i}\right|\cdot s+\left|\frac{1}{2w_i}\right|\cdot(q+d)= \tag{22}$$

$$=\left|\frac{1}{2w_i}\right|\cdot(c+s+q+d)-c-r$$

Як відомо, якщо функція $y=f(x)$ зображена своїм графіком – кривою в декартових координатах, то $f'(x)=tg\alpha$, де α – кут між віссю OX і дотичною до кривої в даній її точці.

Звідси випливає, що $\frac{dT_i}{dH(n)}=tg\alpha_i$, де α_i – кут нахилу прямої $T_i(n,w,w_i)$ до осі OX.

Крім того, з аналізу графіків залежності часу виконання алгоритмів модулярного експоненціювання випливає, що абсолютно стійким є той алгоритм, час виконання якого є константним, тобто пряма зображення якого паралельна до осі абсцис.

Звідси випливає, що стійкішим до часової атаки є той алгоритм, для якого кут нахилу кривої $T(n,w,w_i)$ часу його виконання наближається до 0°. Як відомо, $cos\,0^\circ=1$. Тому для оцінки нормованої стійкості алгоритму можна використати таке співвідношення:

$$S=cos\left(arctg\frac{dT_i}{dH(n)}\right) \tag{23}$$

Аналізуючи рівності (18) – (22), можна зробити висновок, що найвищу стійкість до часового аналізу забезпечує алгоритм β-арного методу "зліва направо".

В таблиці 3 наведено оцінки параметрів кожного алгоритму при $w=4$ та $w_i=3$.

Аналіз таблиці 3 показує, що абсолютно стійким до часової атаки (як і випливало з аналітичного співвідношення) є β-арний метод "зліва направо". Наступним за стійкістю є метод ковзаючого вікна "зліва направо". Найменшу стійкість до часового аналізу має

бінарний метод.

При заданих параметрах найшвидшим алгоритмом модулярного експоненціювання є β-арний метод "справа наліво", який виконується за 3956 такти при $n = 512$, а при довжині $n = 4096$ – за 28150 тактів.

Отже, для побудови сучасних ефективних систем захисту інформації на основі асиметричних криптоалгоритмів найкраще застосовувати реалізацію алгоритму β-арного методу "зліва направо" чи "справа наліво".

Таблиця 3 – Результати досліджень

Алгоритм	Довжина n, Біт	Час виконання, такти	Нормована стійкість S
Бінарний метод "зліва направо"	512	11780	0.062
	1024	23550	
	2048	47110	
	4096	94210	
Бінарний метод "справа наліво"	512	11780	0.062
	1024	23560	
	2048	47110	
	4096	94210	
B-арний метод "зліва направо"	512	4724	1
	1024	9204	
	2048	18160	
	4096	36080	
B-арний метод "справа наліво"	512	3956	0.243
	1024	7412	
	2048	14320	
	4096	28150	
Метод ковзаючого вікна "зліва направо "	512	10970	0.430
	1024	22350	
	2048	44570	
	4096	89020	
Метод ковзаючого вікна "справа наліво"	512	12290	0.114
	1024	21560	
	2048	42590	
	4096	84640	

Висновки

В результаті проведених досліджень алгоритмів модулярного експоненціювання бінарного, β-арного методів та методу ковзаючого вікна, отримані такі результати:

1) побудовано математичні моделі часу виконання кожного з розглянутих алгоритмів, що дало змогу провести детальний аналіз їх продуктивності та стійкості до атак спеціального виду, а зокрема часового аналізу;

2) на основі аналізу цих моделей досліджено продуктивність кожного з методів, що дало змогу виявити алгоритм з найбільшим рівнем продуктивності. Найвищу швидкодію має алгоритм β-арного методу;

3) проведено аналіз залежності часу виконання алгоритму β-арного методу від значення степеня основи, який показав, що для кожного значення довжини n можна знайти таке оптимальне w, при якому відбувається найменша затримка, що дає змогу проектувати високопродуктивні засоби, які реалізують операції сучасних асиметричних криптоалгоритмів;

4) досліджено залежність швидкодії кожного методу від ваги Хемінга; доведено, що при певних заданих умовах найвищу стійкість системи забезпечує алгоритм β-арного

методу "зліва направо", що дає змогу грунтовно обирати алгоритм та його параметри для реалізації засобів криптографічного захисту інформації, стійких до атак спеціальних впливів;

5) наведено результати досліджень значення нормованої стійкості та швидкодії кожного з досліджуваних методів для заданих параметрів, що можуть бути використанні при побудові сучасних високопродуктивних засобів захисту інформації, стійких до атак спеціального виду, що базуються на асиметричних криптографічних алгоритмах.

З отриманих результатів можна зробити висновок, що для забезпечення стійкості до часового аналізу та найвищої швидкодії асиметричної криптосистеми найкраще використовувати алгоритм β-арного методу "зліва направо" чи "справа наліво".

Дослідження решти алгоритмів модулярного експоненціювання проводяться аналогічно.

Перспективним є виділення зі всіх відомих методів той, який буде забезпечувати найвищу швидкодію та стійкість криптосистеми не тільки до часового аналізу, а й до інших атак спеціального виду.

Список літератури

1.*Вербіцький О.В.* Вступ до криптографії. – Львів: Видавництво науково-технічної літератури, 1998. – 247 с.

2.*Ємець В., Мельник А., Попович Р.* Сучасна криптографія. Основні поняття. – Львів: БаК, 2003. –144 с.

3.*Молдовян А.А., Молдовян В.А.,* и др. Криптография. – Серия "Учебники для вузов. Специальная литература". – Спб.: Издательство "Лань", 2000. – 224 с.

4.*Столлингс В.* Криптография и защита сетей: принципы и практика, 2-е изд.: Пер. с англ. – М.: Изд. Дом «Вильямс», 2001 – 672 с.

5.*Чмора А.Л.* Современная прикладная криптография. 2-е изд., стер. – М.: Гелиос АРВ, 2002. – 256 с.

6.*Горпенюк А.Я., Дудикевич В.Б., Ломницький І.Б.* Підвищення швидкодії при обчисленні важкооборотних функцій в асиметричних алгоритмах шифрування. // Науково-технічний журнал "Захист інформації". – 2003 – №1(18) – С.36-43.

7.Seong-Min Hong, Sang-Yeop Oh, Hyunsoo Yoon. New Modular Multiplication Algorithms for Fast Modular Exponentiation. – Spring-Verlag, 1998.

8.*Зайчук А.В.* Основные пути утечки информации и несанкионированного доступа в корпоративных сетях. // Науково-технічний журнал "Захист інформації" – 2003 – № 4. – С. 19-24.

9.*Чеховский С.А., Рудаков Ю.М.* Побочные излучения и защита информации в локальных сетях. // Науково-технічний журнал "Захист інформації" – 2003 – № 4. – С. 30-38.

10. Biham E. and Shamir A. Differential Cryptanalysis of the Data Encryption // Advances in Cryptology -CRYPTO '93 Standard, Springer-Verlag, 1993.

11. Biham E. and Shamir A. Differential Fault Analysis of Secret Key Cryptosystems. In B.Kaliski, editor // Advances in Cryptology -CRYPTO '97,volume 1294 of LNCS , pages 513 – 525. Springer-Verlag, 1997.

12. Muir J. Techniques of Side Channel Cryptanalysis. // A thesis presented to the University of Waterloo in fulfillment of the thesis requirement for the degree of Master of Mathematics in Combinatorics and Optimization, Waterloo, Ontario, Canada, 2001.

13. Ding C., The Differential Cryptanalysis and Design of Natural Stream Ciphers. In *Fast Software Encryption*, Cambridge Security Workshop, December 1993, pages 101-115, Springer-Verlag, Berlin, 1994

14. Vasyltsov I., Vasylkiv L., Vasylkiv N., Chyrka M. Investigation of Modern Exponentiation Algorythms // Proceedings of the International Conference TCSET'2004: "MODERN PROBLEMS OF RADIO ENGINEERING, TELECOMMUNICATIONS AND COMPUTER SCIENCE", – February 24-28, 2004, – Lviv – Slavsko, Ukraine – P. 291-293.

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