1. Предложенный подход позволяет уменьшить дисперсию гистограммы случайной величины, что повышает точность статистической оценки функции распределения.

2. Все исследованные генераторы удовлетворяют критерию χ^2, который для уровня значимости $\alpha = 0.99$ и числа степеней свободы 16 равен 5,812 [11].

3. При использовании заданного уровня значимости критерий хи-квадрат оказывается нечувствительным к коррелированности выборок менее 1/4 их длины. Однако его можно использовать для относительных оценок коррелированности ДВП на меньших промежутках, поскольку по мере увеличения коррелированности его значения увеличиваются.

Литература 1. Зегжда Д. П., Ивашко А. М. Основы безопасности информационных систем.- М.: Горячая линия - Телеком, 2000. - 452 с., ил. 2. Д. Кнут. Искусство программирования для ЭВМ.- Т. 2. - Получисленные алгоритмы. -М.: Мир. - 1977.-482 с. 3. P. L'Ecuyer. Uniform random number generation. // Annals of Operations Research. *-1994.- V. 53. - pp. 77-120. 4. H. Niederreiter. Pseudo-random numbers and optimal coefficients. //* Advances in Mathematics.*-1977.-V. 26. - pp. 99-181. 5. S. K. Park and K. W. Miller. Random number generators: good ones are hard to find. //* Communs of the ACM. *-1988.-V. 31. - pp. 1192-1201. 6. N. S. Altman. Bit-wise behavior of random number generators. //* SIAM Journal of Sci. Stat. Computing.*-1988.-V 9(5) . - pp. 941-949. 7. J. Eichenauer-Herrmann. Inversive congruential pseudorandom numbers: a tutorial. //* International Statistical Review.-1992.-V. 60. - pp. *167-176. 8. J. Walker. HotBits: Genuine random numbers, generated by radioactive decay. // http://www.fourmilab.ch/hotbits/ 9. J. Eichenauer-Herrmann. Statistical independence of a new class of inversive congruential pseudorandom numbers. //* Math. Comp.-1993.-V. 60. - pp. *375-384. 10. I. J. Good. The serial test for sampling numbers and other tests for randomness. //* Proceedings of Cambridge Philosophical Society.*-1953.-V. 49. - pp. 276-284. 11. Справочник по математике (для научных работников и инженеров). - Г. Корн, Т. Корн. -Изд. 4.-М.: Наука, 1978.- 831 с.*

УДК 681.142.35

ВИКОРИСТАННЯ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ ДЛЯ КРИПТОАНАЛІЗУ АСИМЕТРИЧНИХ СИСТЕМ ШИФРУВАННЯ ЕЛЬ-ГАМАЛА ТА RSA

Микола Карпінський, Ярослав Кінах
Тернопільська академія народного господарства

Анотація: **Розглянуто впровадження ідеї паралелізму для алгоритму решета загального числового поля, що дозволить підбирати надійний ключовий матеріал для асиметричних систем шифрування RSA та Ель-Гамала.**

Summary: **This article deals with the RSA encryption algorithm. Its safety is analysed using the number field sieve method. The algorithm work results allow to define a define a secret key in a simple way.**

Ключові слова: **Паралельні обчислення, метод Гауса, надійність системи шифрування.**

Інформаційна безпека є невід'ємною частиною політичної, економічної, оборонної та інших складових національної безпеки. Об'єктами інформаційної безпеки є інформаційні ресурси, канали інформаційного обміну і телекомунікації, механізми забезпечення функціонування телекомунікаційних систем і мереж та інші елементи інформаційної інфраструктури країни. Результатом виконання Концепції національної програми інформатизації є: комплект нормативних документів з усіх аспектів використання засобів обчислювальної техніки для оброблення та зберігання інформації обмеженого доступу; комплекс державних стандартів із документування, супроводження, використання, сертифікаційних випробувань програмних засобів захисту інформації; банк засобів діагностики, локалізації та профілактики вірусів, нові технології захисту інформації з використанням спектральних методів, високо надійні криптографічні методи захисту інформації тощо.

Найбільш перспективним і динамічним напрямком збільшення швидкості розв'язання прикладних задач є широке впровадження ідей паралелізму в роботу обчислювальних систем. Сьогодні спроектовані і випробувані сотні різноманітних комп'ютерів, що використовують у своїй архітектурі той або інший вид

паралельної обробки даних. У науковій літературі і технічній документації можна знайти більше десятка різноманітних назв, що характеризують лише загальні принципи функціонування паралельних машин: векторно-конвеєрні, масивно-паралельні, комп'ютери із широким командним словом, систолічні масиви, гіперкуби, спеціальні процесори і мультипроцесори, ієрархічні і кластерні комп'ютери, dataflow, матричні ЕОМ і багато інших.

Гігантська продуктивність комп'ютерів, що виконують паралельні обчислювання, компенсується складностями їхнього використання. Це твердження відображає закон Амдала [1].

Припустимо, що у програмі відносна кількість операцій, які потрібно виконувати послідовно, дорівнює f, де $0 \le f \le 1$. При цьому відносна кількість операцій обчислюється не за статичною кількістю рядків коду, а за кількістю операцій у процесі виконання.

Крайні випадки в значеннях f відповідають цілком паралельним (f=0) і цілком послідовним (f=1) програмам. Для оцінки прискорення S, яке можна отримати на комп'ютері з p процесорів для даного значення f, можна скористатися законом Амдала:

$$S \le \frac{1}{f + \frac{1-f}{p}}. \qquad (1)$$

Якщо 9/10 програми здійснюється паралельно, а 1/10 як і раніше послідовно, то більшого, ніж у 10 разів, прискорення одержати в принципі неможливо без втрат якості реалізації паралельної частини коду та кількості використовуваних процесорів. Очевидно, що 10-кратне прискорення досягається тільки в тому випадку, коли час виконання паралельної частини дорівнює нулю.

Подивимося на задачу з іншого боку: яку ж частину коду треба прискорити, а значить і попередньо досліджувати, щоб одержати задане прискорення. Використаємо закон Амдала: для того щоб прискорити виконання програми в q разів необхідно прискорити не менше, ніж $\left(1-\frac{1}{q}\right)$ – ту частину програми. Отже, якщо потрібно прискорити програму в 100 разів у порівнянні з її послідовним варіантом, то необхідно одержати прискорення не менше, ніж на 99,99 % коду, що становить значну частину програми.

Звідси перший висновок - перед тим, як переробляти код для переходу на паралельний комп'ютер, треба ґрунтовно проаналізувати код програми. Якщо, оцінивши закладений у програмі алгоритм, зрозуміло, що частка послідовних операцій велика, то прискорення буде незначним. Тоді потрібно замінювати окремі компоненти алгоритму.

Одна з основних вимог, що ставиться до будь-якої системи шифрування - це надійність (в обчислювальному сенсі). Надійність систем шифрування RSA і Ель-Гамаля можна оцінити на основі використання перспективного методу решета загального числового поля, що розпаралелюється на блоки і кожен блок реалізується на окремому комп'ютері. Таке розпаралелення поставленої задачі апріорі значно знизить рівень надійності згаданих систем шифрування, дозволить однозначно обчислювати таємні ключі як відкритими, на практиці використовувати надійний ключовий матеріал, що забезпечить інформаційну безпеку комп'ютерних мереж. Розпаралелення методу решета загального числового поля дозволить проводити криптоаналіз повідомлень користувачів не за рахунок підвищення продуктивності окремого комп'ютера, а завдяки масовому розповсюдженню персональних комп'ютерів і підключення їх до комп'ютерних мереж, зокрема до мережі Інтернет. Найбільш трудоємною частиною методу решета загального числового поля є обробка матриці методом Гауса, що формується під час процесу реалізації методу решета

ISBN 966-622-022-9
Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, вип. 2, 2001 р.

загального числового поля. Тому доцільно значну частину досліджень спрямувати на зменшення затрат обробки вищезгаданої матриці.

Для оцінки ефективності розпаралелення обчислень на N процесорів методу решета загального числового поля використаємо коефіцієнт прискорення алгоритму S_{Np} [2, 3]:

$$S_{Np} = \frac{\xi_{Np}}{\xi_{1p}}, \tag{2}$$

Тут

$$\xi_{Np} = \frac{N^3}{k_1 N^5 - k_2 N^5 + k_3 N^4 - k_4 N^3 + k_5 N^2 - k_6 N + 1}, \tag{3}$$

де $k_1 = 8\alpha^3$; $k_2 = 24\alpha^3$; $k_3 = 3\alpha^2(8\alpha + 1)$; $k_4 = 2\alpha^2(4\alpha + 3)$; $k_5 = 3\alpha(\alpha + 1)$; $k_6 = 3\alpha$; $\alpha = \frac{c_i}{N}$ - коефіцієнт зв'язку матриці; c_i - ширина i-го блокового рядка та i-го блокового стовпця, $i = \overline{1, N}$,

$$\xi_{1p} = \frac{N^2}{k_1 N^5 - k_2 N^4 + k_3 N^3 + k_4 N^2 - k_5 N + 1}, \tag{4}$$

причому $k_1 = 8\alpha^3$; $k_2 = 24\alpha^3$; $k_3 = \alpha^2(24\alpha + 1{,}5)$; $k_4 = -\alpha(8\alpha^2 + 1{,}5\alpha - 1{,}5)$; $k_5 = 1{,}5\alpha$.

На рис. 1 подані графіки залежності функції (2) від кількості процесорів N і коефіцієнту зв'язку α . З ростом числа N прискорення алгоритму S_{Np} стрімко збільшується, після чого зменшується. Тому кожному значенню коефіцієнта α відповідає така кількість підматриць, за якої подальше збільшення кількості процесорів N не дає значного виграшу завдяки блоковому розпаралеленню матричних операцій.

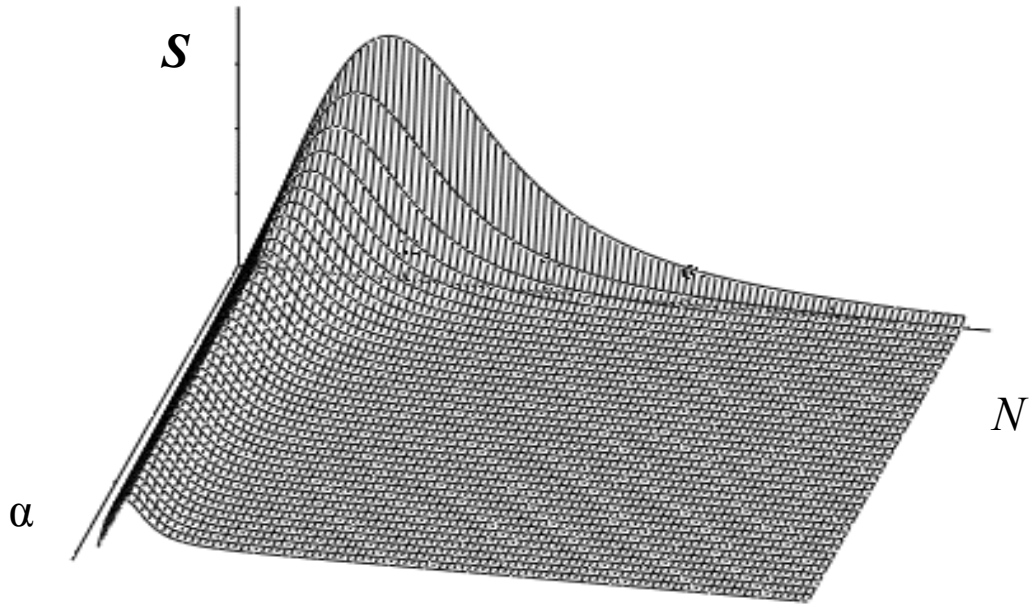

Рисунок 1

Література: 1. Воеводин Вл. В. Курс лекций "Параллельная обработка данных" Online access through WWW: http://www.parallel.ru. 2. Жуков И. Алгоритм решения СЛАУ большой размерности на

ISBN 966-622-022-9
Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні, вип. 2, 2001 р.