RSA Decryption using Earth Simulator

Project Representative
Hidehiko Hasegawa Graduate School of Library, Information and Media Studies, University of Tsukuba

Author
Yasunori Ushiro Department of Mathematics, School of Education, Waseda University

RSA cryptography code is the key technology for safe Internet use and currently a 1,024-bit RSA code is used. To guarantee the safety of RSA code, a decryption time of more than 10 years, even using the fastest supercomputer, is necessary. The present world record for RSA decryption, involving RSA-768 (768 bits), took 1,677 CPU-years to decrypt. So with 1,024-bit RSA code, it is expected to take in the range of 10 to 100 years. All world records for RSA decryption, including that for RSA-768, were carried out by PC clusters. Decryption has never been attempted using a vector supercomputer.

To confirm the decryption time on a vector supercomputer, the author is trying to tune the RSA decryption code for the Earth Simulator 2 (ES2). The RSA cryptography code is based on the difficulty of the factorization of long-digit composite numbers, and the decryption code consists of three parts: "sieve processing", processing of 0-1 matrices, and computation of algebraic square roots. Sieve processing was chosen as the first target for tuning because of its computation time. Sieve processing is tuned, and its performance on one node of the ES2 is approximately 800 times faster than that on a PC (Intel Core 2, 2.3 GHz). This processing is about 99.9% vectorized with few floating point number operations, and it is suitable for the vector supercomputer.

Keywords: RSA code, Sieve processing, Vector processing, non-floating point number operations, GNFS, Decryption

1. Introduction

The RSA cryptography code is the most important key technology for using the Internet safely; however, the currently used 1,024-bit RSA code will no longer be safe to use in near future. The RSA cryptography code is based on the difficulty of the factorization of long-digit composite numbers, and the decryption time of 1,024-bit RSA code is several tens of years even if the fastest supercomputer is used. For an RSA code with a particular number of bits to be safe, its decryption time using the fastest algorithm on the fastest supercomputer must be more than 10 years.

The present world record for RSA decryption, involving RSA-768 (768 bits, 232 digits) was performed by a team consisting of NTT and four foreign organizations in January 2010. It took 1,677 CPU-years to decrypt. This means that if one core of a PC CPU (AMD64, 2.2 GHz) is used, then it will take 1,677 years to decrypt. All reported world records for RSA decryption were carried out using PC clusters; however, there has been no report regarding this challenge for vector supercomputers. Therefore, a test using a vector supercomputer is necessary for the precise evaluation/discussion of the safety of 1,024-bit RSA cryptography codes.

This project intends to obtain basic information for processing RSA decryption on the Earth Simulator 2 (ES2), which is a vector supercomputer. The decryption processing consists of three parts: the first step is "sieve processing", the second step is the processing of 0-1 matrices, and the third step is the computation of algebraic square roots. For RSA-768, the first step, sieve processing, was about 90% of the computation time to decrypt. In 2010, the first year of our project, the author tuned the sieve processing part of the decryption code on the ES2.

2. RSA code

The common key cryptosystem and the public key cryptosystem are basic cryptosystems. The common key cryptosystem has only one key. It is simple and fast to process, but sending the key via the internet represents a problem. The public key cryptosystem has two different keys, one each for encryption and decoding. The key for encryption is open to the public, and the key for decoding can be kept secure because it is not necessary to send this key. A set of keys for the public key cryptosystem is based on the RSA code, which was developed by R. L. Rivest, A. Shamir, and L. M. Adleman in 1978. The RSA code uses two long-digit prime numbers P and Q, and a prime number e to compute $n = P*Q$, $F = (P - 1)*(Q - 1)$, and $D = e^{-1} \pmod F$. Numbers n and e are used as the keys for encryption, and the number D is used for the key for decoding. The safeness of this system is based on the result that, for a given long-digit number n, the factorization algorithm of n to P and Q has high computational complexity and consumes enormous computation time.

3. Sieve method

The sieve method is a factorization method for composite numbers N which obtains a relationship $a^2 - b^2 = 0 \pmod N$ for

Table1 Computational complexity of RSA-768 (232 digits).

	PC-years	Ratio (%)
Exploration of polynomial	20	1
Sieve processing	1500	90
0-1 matrices processing	155	9
Algebraic square root	1	0
Others	1	0
Total	1677	100

some a, b. Because natural numbers a and b are constructed by products of prime numbers provided by the sieve, the exponent of each prime number must be an even number.

For a composite number N, we assume X is the nearest integer to $N^{1/2}$ and calculate $(X + k)^2 - N = A_k$, k = 0, 1, 2, …. Then, we collect A_k that can be factorized using only prime numbers in factor base P. We can factorize N into a product of prime numbers with a combination of A_k whose exponent part is even. This provides the squared form $a^2 - b^2 = 0$ (mod N). Multiple polynomial quadratic sieve (MPQS) uses many types of quadratic equations.

Again for a composite number N, we find a polynomial f(x) and a number M such that f(M) = 0 (mod N). Let θ be an algebraic root of the equation f(x) = 0. We factor a + bM using prime numbers, and factor a + bθ using algebraic elements of primes and the unit. The difference in these factorizations is used for the decryption. For example, let N = 1333, $f(x) = x^3 + 2$, f(M) = N, and M = 11; then 2 + M = 13 and 2 + θ = θ(1 - θ)(1 + θ). Then, 11·(-10)·12=13 (mod 1333) is established. However, we cannot find algebraic elements of primes for any f(x). Therefore, we use a prime ideal in the general number field sieve (GNFS). We define the polynomial norm as N(θ) = |f(-a/b)| for ideal a + bθ, and factorize N(θ) with the prime numbers in the ideal base.

It is said that MPQS is faster for the factorization of fewer than 100 digits, and GNFS is faster for the factorization of more than 100 digits.

For RSA-768, the factorization was carried out using a linear equation and a sixth-order polynomial in the GNFS. The computational complexity of RSA-768 in PC-years of an AMD64 (2.2 GHz) is shown in Table 1.

4. Sieve programming on the ES2

The sieve processing is the most time-consuming part of both MPQS and GNFS. The kernel is as follows:

```
do k=1,N                    N is the number of elements in the base
  do i = Start(k), LP, Prime(k)
                            Start(k): start, Prime(k): increment
    V(i) = V(i) + Log(P(i))
  end do
end do
do i = 1,LP                 < Collection of sieved data >
  if(V(i) .le. PS(i)) then  condition of collection
    ns = ns + 1             ns is the number of collected data
    Sieve(ns) = LLP + i
                            store the position of each collected data
  end if
end do
Update Start(1) through Start(N) for the next sieve
```

Here, the LP is the length for the sieve, Prime(k) is the prime number in the base, and Start(k) is the starting number that it is factorized with prime numbers in the base. For speeding up computation on a PC, LP*4 bytes has to be completely allocated in the cache (1 MB). On the other hand, for the factorization of a 200-digit number, N is at least tens of millions and the value of LP reaches hundreds of millions. To reduce computational complexity, a larger N is necessary but to speed up computation on a PC, a smaller N is necessary. Thus, processing large prime numbers on a PC is very inconvenient.

On the vector supercomputer ES2, a larger LP value can be used and its length becomes the vector length of the ES2. However, for the collection of sieve data part, performance was not good, as expected, because there are almost no calculations. This part was vectorized using a data compaction operation; however, the hit ratio can be once hundreds of millions. The code was modified as follows: first, the existence of adopted data in tens of thousands of intervals was checked, and then the collecting procedure was applied only if the interval had adopted data. Using this modification, the whole sieve process became approximately 3 times faster than the original version on the ES2.

In the sieve processing, the sieve whose loop length is LP, is performed for each prime number. As the sieve processing needs less communication in parallel computing environments, it is easy to parallelize by using MPI.

Table 2 Specification of PC and vector computer.

	PC	Vector computer
Computer	Dell Vostro 200 Intel Core 2 2.3 GHz, 2GB	Earth Simulator (ES2) 3.2 GHz 1 node: 819 Gflops,128 GB
Measurement	1 core Measured by CPU time	1 node (8 CPU) Measured by use time
Software	Windows Vista, g77 -O3 Option	NEC SUPER-UX Auto vector FORTRAN + MPI

5. Comparison of the sieve processing

The specifications of the PC and the vector computer are listed in Table 2.

The author measured the sieve processing for 45- and 60-digit numbers. This is equivalent to the sieve processing of 90 digits and 120 digits in MPQS, and the sieve processing of approximately 130 digits and 170 digits in GNFS. The author used N prime numbers in the base, in ascending order. The computation time greatly depends on the number of elements in the base. The size of LP is 512K on a PC, and 1G on the ES2. Figure 1 shows the result for 45 digits, and Fig. 2 shows that for 60 digits. The computation time on a PC is divided by 200 in both figures.

In both Figs. 1 and 2, the fast range in terms of the number of elements in the base is wider and the numbers are larger on the ES2 than those for the PC. This means that the better performance has been attained on the ES2 for most cases.

It is necessary to include more primes in the base as the number of digits of the factorized number increases. Figure 3 shows the best speed-up ratio of the ES2 over a PC by the number of primes in the base. The left dashed rectangular region is an estimation of 150 digits in MPQS. The speed-up ratio of the ES2 over a PC increases if more primes in the base are used. The ES2 is approximately 600 times faster than the PC for 150 digits of MPQS and is estimated to be approximately 800 times faster for the equivalent RSA decryption size.

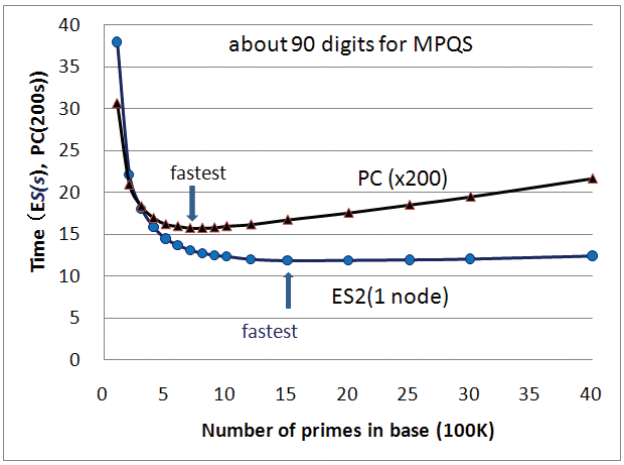

Fig. 1 Sieve processing of 45 digits.

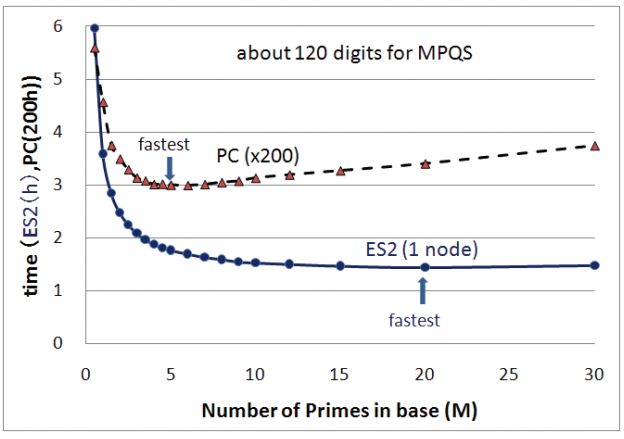

Fig. 2 Sieve processing of 60 digits.

6. Summary

The author tuned the most time-consuming part, sieve processing, in the RSA decryption processing. The following basic information was obtained:

1) This processing is about 99.9% vectorized with few floating point number operations, and it is suitable for a vector supercomputer.
2) The performance of the sieve processing on one node of the ES2 is approximately from 200 to 800 times faster than that of a PC (Intel Core 2, 2.3 GHz).
3) The performance ratio for a realistically scaled RSA decryption is expected to be approximately 800.

For the second year of this study, the author will modify the sieve processing in the GNFS, measure its computation time for from 100 digits to 200 digits, progressively, and then estimate the computation time of GNFS for more than 200 digits. The author will also tune the 0-1 matrices processing.

Acknowledgement

The author expresses his thanks to Dr. Yoshinari Fukui and Dr. Toshiyuki Asano of JAMSTEC, who provided vital suggestions for greatly speeding up computation on the ES2.

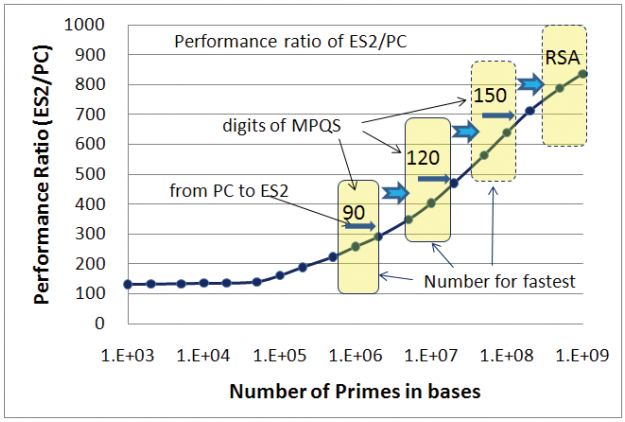

Fig. 3 Speed-up ratio of the ES2 over a PC.

References

[1] Richard A. Mollin, "RSA and PUBLIC-KEY CRYPTOGRAPHY", Chapman and Hall/CRC, 2002.

[2] Yuji Kida, "Prime factoring by General Number Field Sieve", 2003.
http://www.rkmath.rikkyo.ac.jp/~kida/nfs_intro.pdf (in Japanese)

[3] Jun'ichi Yamazaki and Souta Kamaike, "The 2010 problem of the Cryptography", ASCII technologies, 75-93, Sept. 2010. (in Japanese)

[4] Neal Koblitz, translated by Kouichi Sakurai, "A Course in Number Theory and Cryptography ", Springer, 1997. (in Japanese)

[5] Kazumaro Aoki, "Advances in Integer Factoring Technique: The Way to Factor RSA-768", IPSJ Magazine, 51(8), 1030-1038, Aug. 2010. (in Japanese)

[6] Yasunori Ushiro, "A high speed sieve computation for the RSA decryption by the vector computer", RIMS kokyuroku, No.1733, 101-117, March 2011. (in Japanese)

地球シミュレータを用いたRSA暗号解読処理

プロジェクト責任者
長谷川秀彦　　筑波大学大学院　図書館情報メディア研究科
著者
後　　保範　　早稲田大学　教育学部　数学科

ES2 において RSA 暗号解読の大半の時間を占める「ふるい処理」の高速化を行い、ES2 1 ノードで PC 1 コアの約 800 倍の性能を達成した。

RSA 暗号はインターネットを安全に使ううえで欠かせない技術である。しかし、現在使用している 1024 ビットの RSA 暗号が安全性の問題で近いうちに使えなくなるという「2010 年問題」が懸念されている。RSA 暗号には桁数の多い合成数の因数分解の困難性が利用されており、1024 ビット RSA 暗号の安全性はスーパーコンピュータを数年使用しても解読されないという仮定のもとで成り立っている。いっぽう、2010 年 1 月に NTT 他 4 カ国の共同で実施された RSA 暗号解読の世界記録（RSA-768）をはじめ、今までのすべての世界記録は PC クラスタで達成されており、ベクトル方式のスーパーコンピュータによる RSA 暗号の解読実験は全く報告されていない。そこで、代表的なベクトル方式のスーパーコンピュータである地球シミュレータ（ES2）において RSA 暗号の解読実験を行う。

一般的な RSA 暗号解読は、ふるい処理、0-1 行列処理、代数的平方根の計算の 3 段階からなり、これまでの RSA 暗号解読プログラムはすべて PC 用となっている。本年度は、RSA 暗号解読の大半の処理時間を占める「ふるい処理」と ES2 の相性を評価した。PC クラスタにおける RSA-768 ではふるい処理が全体の約 90% を占めている。試行錯誤の結果、ES2 で効率よくふるい処理を行うにはふるい結果のデータを集める処理の対策が重要であることが判明した。これらは PC における対策とは大幅に異なっているが、ES2 とふるい処理の相性は良いことが分かった。数値実験では、800 bit 相当のふるい処理において ES2 の 1 ノードで 2.3GHz の PC 1 コアの 800 倍程度の性能を達成した。このとき、浮動小数演算はほとんどないがベクトル化率は 99.9% で、この結果を単純に外挿すると ES2 32 ノード、2TB（64GB/ ノード）を用いた RSA-768 のふるい処理に要する時間は 21 日となる。

本年度は、0-1 行列処理の部分を ES2 向けに書き換えること、ふるい処理と 0-1 行列処理を合わせて 1024 bit の RSA 暗号の解読に要する時間を推定するための基礎的データの取得を行う予定である。

キーワード:RSA暗号, ふるい処理, ベクトル処理, 整数演算, GNFS

謝辞

地球シミュレータ（ES2）における高速化について、貴重なご意見を頂いた海洋研究開発機構地球シミュレータセンターの福井義成氏及び浅野俊幸氏に謹んで感謝の意を表する。

参考文献

[1] Richard A. Mollin, "RSA and PUBLIC-KEY CRYPTOGRAPHY", Chapman and Hall/CRC , 2002 年
[2] 木田　祐司 , " 数体ふるい法による素因数分解 ", 2003 年 http://www.rkmath.rikkyo.ac.jp/~kida/nfs_intro.pdf
[3] 山崎　潤一 , 釜池 聡太 , " 暗号の 2010 年問題 ", ASC Ⅱ Technologies, 75-93,2010 年 9 月
[4] N. コブリッツ , 桜井 幸一訳 ," 数論アルゴリズムと楕円暗号理論入門 ", シュプリンガー・フェアラーク東京 , 1997 年
[5] 青木和磨呂 , " 素因数分解技術の進展：RSA-768 の分解達成への道のり ", 情報処理 , 51(8), 1030-1038, 2010 年 8 月
[6] 後　　保範 , " ベクトル計算機による RSA 暗号ふるいの高速化 ", 京大 数理研 講究録 , No. 1733, 101-117, 2011 年 3 月