

# Cryptographic and Computational Challenges in Grid Computing

Song Y Yan & Glyn James

School of Mathematical and Information Sciences, Coventry University, Coventry CV1 5FB, UK  
s.yan@coventry.ac.uk

Gongyi Wu

College of Technical Information Science, Nankai University, Tianjin 300071, China

## Abstract

*A computational grid is a large-scale distributed computing environment capable of providing dependable, consistent, pervasive, and inexpensive access to high-end computational resources so that complicated computation tasks, such as the verification of the Riemann hypothesis, the factorization of large composite numbers, and the break of difficult RSA codes, can be performed in a relatively cheap distributed environment rather than an expensive supercomputer. As different resources in a grid may have different access policies and different security measures, it is usually more difficult to achieve the secure computation in a distributed grid environment than a centralized computing environment. In this paper, we should discuss some security and computation challenges in the grid computing environment, with an emphasis on encryption and authentication via zetaGrid, a grid computing environment for verification of the Riemann hypothesis.*

## 1. Security Services for Grid Computing

A computational grid is a large-scale distributed computing environment capable of providing dependable, consistent, pervasive, and inexpensive access to high-end computational resources so that complicated tasks such as the verification of the Riemann hypothesis can be performed in a relatively cheap distributed environment rather than an expensive supercomputer. As different resources in a grid may have different access policies, a computational grid should provide its users a good security service, including but not limited to: 1) authentication, 2) authorization/access Control, 3) credential conversion, 4) identity mapping, 5) confidentiality/privacy, and 6) nonrepudiation, secure computation.

Current grid security service is based on the combined use of PKI, XML and SSH, etc. We argue that for a grid to be more useful and realistic a much higher level of security service will be needed. In what follows, we consider three extreme cases: 1) secure computation: Suppose user  $A$  wants to use resource  $R$  of organization  $O$  to perform a task  $C$ , but  $A$  does not want  $O$  to know anything about  $C$ . To stop  $O$  to understand  $C$ , all information involved in the computation is encrypted, and even all operations are performed on the encrypted information. Upon completing the computation,  $A$  gets back the result and decrypt it at his end. 2) confidentiality: Organization  $O$  may only allow user  $A$  to use its Resource  $R$ , but does not allow  $A$  to access  $O$ 's any other resources or information. At the  $O$ 's side, only the allowed resources and information are open to the grid users, any other resources are blocked and protected by firewalls and any sensitive information are encrypted. 3) cryptovirology: Suppose user  $A$  has an access to organization  $O$ 's resource  $R$ , but user  $B$  does not have. It may, however, well be possible that user  $B$  can get access to  $R$ , more seriously,  $B$  can produce/spread virus on  $R$  and hence the whole grid. According to the Church-Turing thesis, there is not a general program to detect whether or not a resource is infected by a virus. A way to reduce the risk of infection of virus is to use kleptography (a method of using cryptography against cryptography).

As can be seen from the above discussion that cryptography, particularly public-key cryptography (PKC) is an essential operation in almost all the security cases. It is interesting to note that almost all the *practical* public-key cryptosystems in use are based their security on one of the following three computationally infeasible mathematical problems: 1) *The integer factorization problem (IFP)*: Given  $N \in \mathbb{Z}_{>1}$ , find a factor  $f$  of  $N$  such that  $f \mid N$ . 2) *Discrete logarithm problem (DLP)*: Given  $x, y, N$ , find  $k$ , such that

$y \equiv x^k \pmod{N}$ . 3) *The elliptic curve discrete logarithm problem (ECDLP)*: Given  $P, Q, N$ , find  $k$ , such that  $Q \equiv kP \pmod{N}$ , where  $P, Q$  are point on an elliptic curve  $y^2 \equiv x^3 + bx + c \pmod{N}$ . The fastest algorithm for IFP and DLP is the Number Field Sieve (NFS), a variant of the Index calculus, runs in sub-exponential time and hence is inefficient.

We list in the following **two Computation Challenges for Grid Computing**:

1. Factor the following 212 digits (704 bits) number:  
 74037563479561712828046796097429573142593188889\_23128908493623263897276503402826627689199641962\_51178439958943305021275853701189680982867331732\_73108930900552505116877063299072396380786710086\_096962537934650563796359,

2. Factor the following 617 digits (2048 bits) number:  
 25195908475657893494027183240048398571429282126\_20403202777713783604366202070759555626401852588\_07844069182906412495150821892985591491761845028\_08489120072844992687392807287776735971418347270\_26189637501497182469116507761337985909570009733\_04597488084284017974291006424586918171951187461\_21515172654632282216869987549182422433637259085\_14186546204357679842338718477444792073993423658\_48238242811981638150106748104516603773060562016\_19676256133844143603833904414952634432190114657\_54445417842402092461651572335077870774981712577\_24679629263863563732899121548314381678998850404\_45364023527381951378636564391212010397122822120\_720357

RSA Security Inc will pay \$30,000 and \$200,000 to the first person/group who finds the two factors of these two numbers, respectively. The ECDLP is even harder than IFP/DLP, since there is no sub-exponential time algorithm for it, e.g., the famous Xedni calculus for ECDLP cannot be run in sub-exponential time. So the PKC based on IFP/DLP/ECDLP is secure at present, but of course, it may not be secure in the future. For grid computing, the most important security threat would be someone who uses cryptography against someone else cryptography. For example, user  $A$  may use his public-key to write a virus program to produce/spread virus over the grid, one-one can detect the virus except the virus writer who has the private key. In this case, the grid will be totally insecure; new technologies such as kleptography will be needed to ensure the grid security.

## 2. Security Services in ZetaGrid

In 1859, in an effort to prove the Prime Number Theorem, the great German mathematician Bernhard

Riemann extended the Euler product formula

$$\zeta(s) = \prod_p \left( \frac{1}{1 - p^{-s}} \right)$$

where the product runs over all the prime numbers, to the function of a complex variable [9]:

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^{-s}}$$

where  $s = \sigma + it$  with  $i = \sqrt{-1}$  and  $\sigma, t \in \mathbb{R}$ . Riemann's great insight was to study the  $\zeta$ -function for complex values of  $s$  and showed that  $\zeta(s)$  is analytic for  $\sigma > 1$  and can be continued across the line  $\sigma = 1$  (see Figure 1).

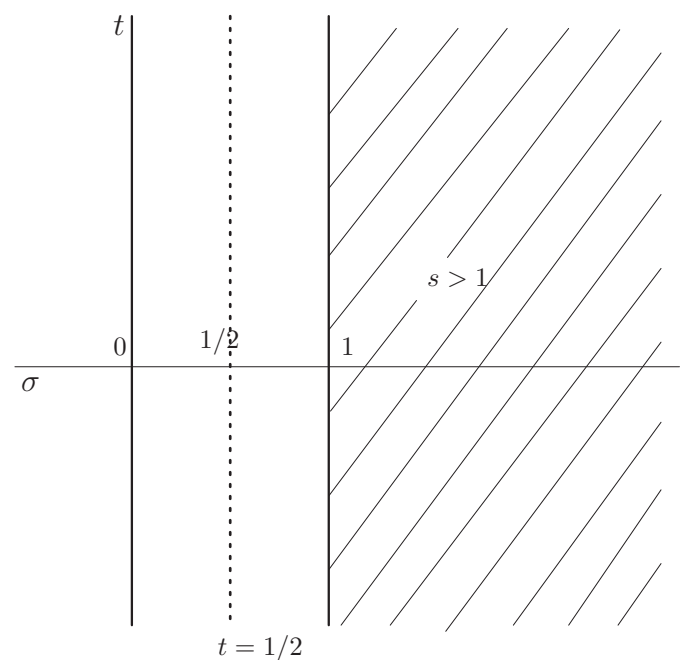


Figure 1. The complex plane of  $\zeta$ -function

It is clear that the  $\zeta$ -function has no zeros in the half-plan for  $\sigma > 1$ , and also no zeros for  $\sigma = 1$  and hence no zeros for  $\sigma = 0$ . Therefore, there are only two types of zeros for  $\zeta(s)$ :

- 1) *Zeros lying outside the critical strip*:  $\zeta(\sigma + it) = 0$  for  $t = 0$  and  $\sigma = -2, -4, -6, -8, -10, \dots$ . These are the real zeros (trivial zeros); there are infinitely many such zeros.
- 2) *Zeros lying in the critical strip*:  $\zeta(\sigma + it) = 0$  for  $0 < \sigma < 1$  and  $t \in \mathbb{R}$ . These zeros are called the complex zeros (non-trivial zeros). There are also infinitely many such zeros. But Riemann conjectured that

**Conjecture 1 (Riemann's hypothesis)**

$$\zeta\left(\frac{1}{2} + it\right) = 0, \quad t \in \mathbb{R}.$$

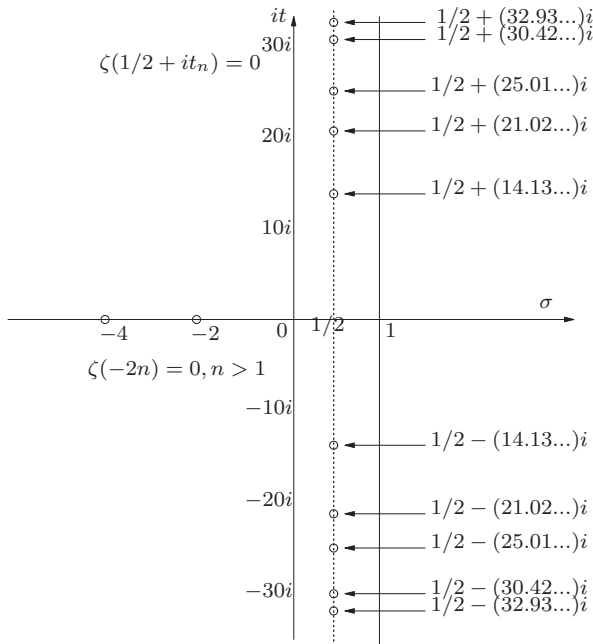


Figure 2. Zeros of the Riemann  $\zeta$ -function

More than 200,000,000,000 complex zeros have been found to date; all of them are indeed lying on the critical line  $\sigma = 1/2$ ). On 24 May 2000, the Clay Mathematical Institute in Boston, listed Riemann's hypothesis as one of its seven millennium prize problems, each with one million US dollars:

**A Millennium Prize Problem:** Prove or disprove Riemann's hypothesis.

The Riemann Hypothesis (RH) is fundamental to the Prime Number Theorem (PNT) as indicated in the following formulas:

$$\begin{aligned} \pi(x) &\sim \infty && \text{(Euclid, about 250 BC)} \\ &\sim \int_2^x \frac{dt}{\ln t} && \text{(Prime Number Theorem, 1896)} \\ &= \int_2^x \frac{dt}{\ln t} + \mathcal{O}(xe^{-A\sqrt{\ln x}}) && (1899) \\ &\stackrel{?}{=} \int_2^x \frac{dt}{\ln t} + \mathcal{O}(\sqrt{x} \ln x) && \text{(Conjecture)} \\ &\Leftrightarrow \text{Riemann's hypothesis} && (1859) \\ &\Rightarrow \$1,000,000 \text{ (Millennium Prize)} \end{aligned}$$

ZetaGrid (see [6] and [7]) is a platform independent grid system that uses idle CPU cycles from participating computers. It is based at the IBM Development Laboratory in Böblingen and used to verify the Riemann Hypothesis. If we can find a complex zero, that is in the critical strip  $0 < \sigma < 1$ , but not on the critical line  $\sigma = \frac{1}{2}$ , then the Riemann hypothesis is disproved. In any case, the numerical verification of the hypothesis will help mathematicians to have a better understanding of the hypothesis and hence the prime distributions. At present, ZetaGrid runs one more than 11,000 computers/workstations in more than 70 countries and has a peak performance rate of about 7056 GFLOPS. The ZetaGrid security architecture can be shown as follows (see Figure 3): It can be seen from Figure 3 that

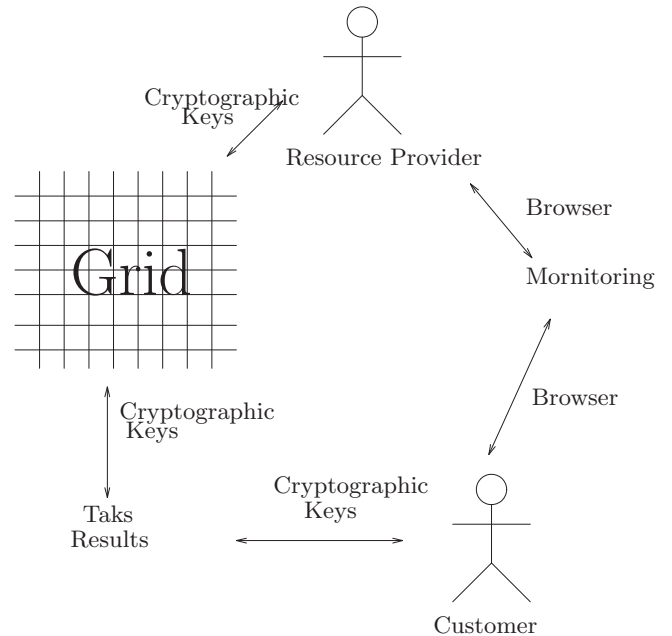


Figure 3. ZetaGrid Security Architecture

cryptography (indicated by keys) plays a central role in the security architecture. It is true that cryptography is essentially the only automated tools in network and information security [4]. However, cryptography itself has weakness, and particularly for efficient encryption/decryption, a great deal of restrictions have been placed on the use (especially the size) of the cryptographic keys. We shall discuss in the next section some practical security problem in RSA cryptography, the most popular and widely used cryptosystem.

**3. Security Challenges in Cryptography**

The RSA cryptosystem [3], invented by Rivest, Shamir and Adleman in 1978, can be formally defined

as follows:

$$\text{RSA} = (\mathcal{M}, \mathcal{C}, \mathcal{K}, M, C, e, d, N, E, D) \quad (1)$$

where

- 1)  $\mathcal{M}$  is the set of plaintexts, called the plaintext space.
- 2)  $\mathcal{C}$  is the set of ciphertexts, called the ciphertext space.
- 3)  $\mathcal{K}$  is the set of keys, called the key space.
- 4)  $M \in \mathcal{M}$  is a piece of particular plaintext.
- 5)  $C \in \mathcal{C}$  is a piece of particular ciphertext.
- 6)  $N = pq$  is the modulus with  $p, q$  prime numbers, usually each with at least 100 digits.
- 7)  $\{(e, N), (d, N)\} \in \mathcal{K}$  with  $e \neq d$  are the encryption and decryption keys, respectively, satisfying

$$ed \equiv 1 \pmod{\phi(N)} \quad (2)$$

where  $\phi(N) = (p-1)(q-1)$  is the Euler  $\phi$ -function and defined by  $\phi(N) = \#(\mathbb{Z}_N^*)$ .

- 8)  $E$  is the encryption function

$$E: \mathcal{M} \xrightarrow{(e, N)} \mathcal{C}$$

which maps  $M \in \mathcal{M}$  to  $C \in \mathcal{C}$ , using the public-key  $(e, N)$ , such that

$$C \equiv M^e \pmod{N}. \quad (3)$$

- 9)  $D$  is the decryption function

$$D: \mathcal{C} \xrightarrow{(d, N)} \mathcal{M}$$

which maps  $C \in \mathcal{C}$  to  $M \in \mathcal{M}$ , using the private-key  $(d, N)$  such that

$$M \equiv C^d \pmod{N}, \quad (4)$$

satisfying

$$M \equiv C^d \equiv (M^e)^d \equiv M \pmod{N}. \quad (5)$$

Note that for an authorized user who knows  $d$ ,  $M \equiv C^d \pmod{N}$  is easy calculate. Of course, knowledge of any one of the following four elements of the trap-door information  $\{d, p, q, \phi(N)\}$  can lead to the decryption of  $C$  easily, since they are polynomially reducible each other, i.e., knowing any one of them can get the rest in polynomial time

$$\{d, p, q, \phi(N)\} \implies \text{RSA}(M).$$

Of course, all these four pieces of trap-door information can be obtained from the factorization of  $N$ , and vice versa:

$$\text{IFP}(N) \iff \{d, p, q, \phi(N)\}.$$

However, for the unauthorized person who does not know the trap-door information will need to invert the RSA function  $M \mapsto M^e \pmod{N}$ , which is conjectured to be as hard as IFP.

**Conjecture 2 (RSA)** *Recovering the RSA plaintext  $M$  from its corresponding ciphertext  $C$  is as hard as factoring  $N$ . That is,*

$$\text{RSA}(M) \iff \text{IFP}(N).$$

Of course, this is just an unproved assumption. What we know at present is that

$$\text{IFP}(N) \stackrel{\vee}{\implies} \text{RSA}(M), \quad \text{IFP}(N) \stackrel{?}{\longleftarrow} \text{RSA}(M).$$

In fact, at least for some cases,  $\text{IFP}(N) \longleftarrow \text{RSA}(M)$  is not true, that is, we can recover  $M$  without factoring  $N$  at least for some cases. Of course, this does not imply that RSA is insecure in general. In what follows, we shall give one such example. Let  $N$  be the 704 bits (212 digits) number mentioned previously, which is out of reach by any computational grid or supercomputer at present:

74037563479561712828046796097429573142593188889\_23128908493623263897276503402826627689199641962\_51178439958943305021275853701189680982867331732\_73108930900552505116877063299072396380786710086\_096962537934650563796359

$M$  the plaintext:

19050321180920250019051822090305190009190013151\_80500091315182001142000200801140009140001142500\_1529080518000315131621200914070019251920051419

which is the digital representation of “security services in grid environment is more important than in any other computing systems” using the coding scheme that space  $\rightarrow 00, a \rightarrow 01, b \rightarrow 02, \dots, y \rightarrow 25, z \rightarrow 26$ . Suppose also that the message was sent twice using  $e_1 = 9007$  and  $e_2 = 65537$  but  $N$  remains the same for the two encryptions, so that  $C_1 \equiv M^{e_1} \pmod{N}$  becomes:

54213758554607200816975921786630083305351106553\_88370798720814878417308955639781370911277031901\_79645941562543942769490767445418996735567151540\_20706435467352298940498099574848803558307012586\_72065368736965583162749

and  $C_2 \equiv M^{e_2} \pmod{N}$  becomes:

30014909197164710381912137017265493576844033959\_45784933018511699958290486566281791406858689013\_90448075431012753974124910283623149731354583104\_93461868489822091555373215221945212927115709165\_922959865942089522763123

As indicated by the RSA conjecture, to recover  $M$  from  $C$  is as hard as factoring  $N$ , but as everybody knows, factor the 704 bits number is out of reach at present. What we will show in the following is that we can recover  $M$  from  $C_1$  and  $C_2$  without any knowledge of the factors of  $N$ . First we solve the bilinear Diophantine equation:

$$9007x + 65537y = 1$$

by obtaining the convergents of the continued fraction of  $9007/65537$ :

$$0, \frac{1}{7}, \frac{3}{22}, \frac{4}{29}, \frac{7}{51}, \frac{7}{51}, \frac{11}{80}, \frac{18}{131}, \frac{29}{211}, \frac{76}{553}, \frac{105}{764}, \frac{181}{1317}, \\ \frac{286}{2081}, \frac{467}{3398}, \frac{1220}{8877}, \frac{9007}{65537}.$$

Then

$$\begin{cases} x = (-1)^{n-1}q_{n-1} = (-1)^{13}8877 = -8877, \\ y = (-1)^np_{n-1} = (-1)^{14}1220 = 1220. \end{cases}$$

Therefore,  $((C_1^{-8877} \pmod{N}) \cdot (C_2^{1220} \pmod{N})) \pmod{N}$  will be:

19050321180920250019051822090305190009190013151\_80500091315182001142000200801140009140001142500\_1529080518000315131621200914070019251920051419,

which is exactly our wanted original plaintext! This example shows that we need to be very careful when using encryption. Of course, this is nothing wrong with the RSA encryption – RSA is a strong and widely used encryption scheme for which its three inventors Ron Rivest, Adi Shamir and Leon Adleman received the Year 2002 Turing Award (A equivalent Nobel Prize for Computer Science), it is also nothing wrong with the modulus  $N$  which is still unfactorable by today's computing facilities including grids and supercomputers. It is in fact with the use of RSA; we need a proper use of RSA and we need a proper understanding of RSA! Any Grid user who wants to securely use their grid needs to have a deep understanding of cryptography and mathematics!

#### 4. Conclusions

In this paper, we have examined some security and computation challenges of grid computation. As Prof

Peter Wegner at Brown University noted: “the increased use of shared communications channels, particularly wireless and local area networks (LAN's), leads to greater connectivity, but also to a much greater opportunity to intercept data and forge messages, ... The only practical way to maintain privacy and integrity of information is by using public-key cryptography”. This interesting remark also fits the grid computation very well. To make a secure grid computing environment, we not only need to develop new strong encryption schemes, but also need to use the existing encryption schemes properly and intelligently.

#### References

- [1] E. Bombieri, Problems of the Millennium: The Riemann Hypothesis, Clay Mathematics Institute, Boston, 2001.
- [2] I. Foster and C. Kesselman (Editors), *Grid - Blueprint for a New Computing Infrastructure*, 2nd Edition, Morgan Kaufmann, 2004
- [3] R. L. Rivest, A. Shamir and L. Adleman, A Method for Obtaining Digital Signatures and Public Key Cryptosystems, *Communications of the ACM*, **21**, 2(1978), 120–126.
- [4] W. Stallings, *Cryptography and Network Security*, 3rd Edition, Prentice-Hall, 2003.
- [5] E. C. Titchmarsh, *The Theory of the Riemann Zeta-Function*, 2nd Edition (Revised by R. D. Heath-Brown), Oxford University Press, Oxford, 1986.
- [6] S. Wedeniwski, ZetaGrid - Experiences with the Grid for everybody, *Grid Computing News*, Ehnningen, Germany, November 2003.
- [7] S. Wedeniwski, ZetaGrid - Computational verification of the Riemann Hypothesis, *Conference in Number Theory in Honour of Professor H.C. Williams*, Banff, Alberta, Canada, May 2003.
- [8] V. Welch, and F. Siebenlist, et al., Security for Grid Services, Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing (HPDC'03), IEEE Computer Society, 2003, 48-57.
- [9] S. Y. Yan, *Number Theory for Computing*, 2nd Edition, Springer-Verlag, 2002.