Performance Analysis of Elliptic Curve Cryptography Using Onion Routing to Enhance the Privacy and Anonymity in Grid Computing

H. Parveen Begam and Maluk Mohamed M. A.

***Abstract*—Grid computing is an environment that allows sharing and coordinated use of diverse resources in dynamic, heterogeneous and distributed environment using Virtual Organization (VO). Security is a critical issue due to the open nature of the wireless channels in the grid computing which requires the three fundamental services: authentication, authorization and encryption. The privacy and anonymity are considered as an important factor, while communicating over publicly spanned network like web. To ensure a high level of security we explored an extension of onion routing, which has been used with dynamic token exchange along with protection of privacy and anonymity of individual identity. To improve the performance of encrypting the layers, the Elliptic Curve Cryptography is used. Compared to traditional cryptosystems like RSA (Rivest-Shamir-Adelman), ECC (Elliptic Curve Cryptosystem) offers equivalent security with smaller key sizes, which results in faster computations; lower power consumption, as well as memory and bandwidth savings. This paper presents the estimation of the performance improvements of onion routing using ECC as well as the comparison graph between performance level of RSA and ECC.**

***Index Terms*—Anonymity, ECC, grid computing, onion routing, privacy, RSA.**

I. Introduction

Grid computing system has emerged as a special form of distributed computing and is distinguished from conventional distributed computing systems by its focus on larger-scale resource sharing. As the Grid approach was widely discussed and experimented, the objectives of the Grid computing has been generalized to refer to large-scale sharing of resource, storage or other expensive equipments, over a wide geographical distribution. With the rapid development of the global information infrastructure, the use of virtual organization (VO) is gaining increasing importance as a model for studying business and organizational structures. To further enhance the pervasiveness of VO, it is of great importance that participation of mobile computing nodes be supported. Thus, security is a critical issue due to the open nature of the wireless channels that provide connectivity to mobile devices. An extension of onion routing with dynamic token exchange for protecting from intruders within service oriented computational grid backbone. RSA is the most commonly used public-key cryptosystem today. The security of a system is only as good as that of its weakest component; for this reason, the work factor needed to break a symmetric key must match that needed to break the public-key cryptosystem used for key establishment. Due to expected advances in cryptanalysis and increases in computing power available to an adversary, both symmetric and public-key sizes must grow over time to offer acceptable security for a fixed protection life span, and the Elliptic Curve Cryptosystem (ECC), offers the highest strength per bit of any known public-key cryptosystem today. ECC not only uses smaller keys for equivalent strength compared to traditional public- key cryptosystems like RSA, the key size disparity grows as security needs increase. This makes it especially attractive for constrained wireless devices because smaller keys result in power, bandwidth and computational savings. So mutual authentication and secure communication is to be presented. In this paper we present the estimation of the performance improvements of onion routing using ECC as well as the comparison graph between performance levels of traditional RSA with ECC.

The remainder of this paper is organized as follows. Section 2 derives the related work. Section 3 derives the proposed work along with the information about ECC algorithm implementation, secure data transfer using onion routing. Section 4 derives the comparison of ECC and RSA. Section 5 presents conclusions and future directions.

Manuscript received April 10, 2012; revised May 5, 2012

The authors are with Software System Group, Department of Computer Science and Engg, M.A.M. College of Engineering, Anna University Tiruchirappalli, India (e-mail: ssg_parveen@mamce.org, ssg_malukmd@mamce.org).

II. Related Work

With the rapid development of the global information infrastructure, the use of virtual organization (VO) is gaining increasing importance as a model for studying business and organizational structures. The notion of VO is significant in that it could serve as a basic framework for implementing geographically distributed, cross-organizational application systems in a highly flexible manner. To further enhance the pervasiveness of VO, it is of great importance that participation of mobile computing nodes be supported. Thus, security is a critical issue due to the open nature of the wireless channels that provide connectivity to mobile devices [1]. It also discusses the design of security infrastructures that support mobile nodes in mission-specific applications. A simple grid security infrastructure that supports participation of mobile computing nodes is also proposed to illustrate the implementation feasibility of the infrastructure.

Privacy of message during transactions between grid nodes belonging to public realm has become a mounting apprehension for nodes engaged in the communication [3].

While it is imperative to ensure a high level of security for geographically dispersed distributed nodes, privacy issues for concealing individual profiles and identities are worth concerning.

Elliptic Curve Cryptography (ECC) is emerging as an attractive public-key cryptosystem for mobile/wireless environments. Compared to traditional cryptosystems like RSA, ECC over's equivalent security with smaller key sizes, which results in faster computations, lower power consumption, as well as memory and bandwidth savings. This is especially useful for mobile devices which are typically limited in terms of their CPU, power and network connectivity. However, the true impact of any public-key cryptosystem can only be evaluated in the context of a security protocol [4]. It presents a first estimate of the performance improvements that can be expected in SSL (Secure Socket Layer), the dominant security protocol on the Web today, by adding ECC support.

III. Proposed Work

Due to the dynamic nature of sharing relationships in VO, security solutions for Grid computing systems must allow applications to coordinate diverse access control policies and to operate securely in heterogeneous environments. Grid security solutions need to provide mutual authentication that allows a user, the processes that comprise a user's computation, and the resources used by those processes, to verify each other's identity.

A. Implementation of ECC and RSA

The core of elliptic curve arithmetic is an operation called scalar point multiplication, which computes Q = kP (a point P multiplied k times resulting in another point Q on the curve). Scalar multiplication is performed through a combination of point-additions (which add two distinct points together) and point-doublings (which add two copies of a point together). For eg, 11P can be expressed as 11P = (2 * ((2 *(2 * P)) + P)) + P. ECC algorithm consists of three steps: Signature generation, Signature verification. The procedural steps for traditional RSA algorithm and for ECC algorithm is as follows.

B. RSA key generation

1. Select random prime numbers p and q, and check that p !=q
2. Compute modulus n = pq
3. Compute phi, $\Phi = (p-1)(q-1)$
4. Select public exponent e, $1 < e < \Phi$ such that $\gcd(e,\Phi) = 1$
5. Compute private exponent $d = e^{-1} \bmod \Phi$
6. Public key is {n, e},private key is {n, d}
7. Encryption: $c = m^e \bmod n$, decryption: $m = c^d \bmod n$
8. Digital signature: $s=H(m)^d \bmod n$, Verification: $m' = s^e \bmod n$,
9. If m' = H(m) signature is correct, H is a publicly known hash function.

C. ECC digital Signature Algorithm (ECDSA)

For signing a message m by sender A, using A's private key d_A

1. Calculate e=HASH(m), where HASH is a cryptographic hash functions, such as RIPEMD,MD5,SHA-1.
2. Select a random integer k from [1,n-1]
3. Calculate r=x1(mod n), where (x1,y1)=k*G. If r=0, go to step 2
4. Calculate $s=k^{-1}$ $(e=d_A r)(\bmod n)$. If s=0, go to step 2
5. The signature is the pair (r,s)

D. ECC Digital Signature Verification

For B to authenticate A's Signature, B must have A's public key Q_A

1. Verify that r and s are integers in [1,n-1]. If not, the signature is invalid
2. Calculate e= HASH(m), where HASH is the same function used in the signature generation
3. Calculate $w=s^{-1} \bmod n$
4. Calculate $u_1=ew(\bmod n)$ and $u_2=rw(\bmod n)$
5. Calculate $(x1,y1)u_1G=u_2Q_A$
6. The signature is valid if x1=r(mod n), invalid otherwise.

RSA is an algorithm for public-key cryptography that is based on the presumed difficulty of factoring large integers, the factoring problem. The user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key. The prime factors must be kept secret. To overcome the drawbacks of RSA, ECC algorithm is generated.

TABLE1: Key Sizes for Various Encryption Methods.

Symmetric	ECC	RSA/DH/DSA
80l	163	1024
128	283	3072
192	409	7680
256	571	15360

The above table shows the various key sizes for symmetric, ECC and RSA algorithms. Comparatively the ECC algorithm takes less number of keys for encryption and decryption. So ECC along with onion routing helps to secure the data at transmission channel.

E. Secure Data Transfer using Onion Routing

The possibility of hacking is more at the transmission channel. To secure the data at transmission channel Onion Routing concept is used. In this architecture the server nodes are divided into edge nodes (En nodes), lying closer to the network edge, and core nodes (Cn node), placed in strategic positions within the grid environment. The edge nodes will finally collate the services from several service providers, while the core nodes are responsible to form the part of onion routing for finding out and finally availing the desired service. Here are the steps.

(1) Designated sender requests for a service.

(2) Service request is received by a designated Edge node. The edge node extracts the identification of the service requestor and generates a dynamic token (ticket) based on internally generated encryption technique similar to

our dynamic token generation algorithm (ECC). It then finds out a core node in the onion routing layer to get the desired service. The edge nodes store the sender's details and then find the suitable core node in an onion routing framework.

(3) The core node then seeks for the subsequent core nodes in onion routing methodology and establishes a path to flow the message to the respective Service Provider or a set of Service Providers. The nodes may have multiple roles of a broker, collator, sender on behalf of some other service requestor etc. The message flow is done in an encrypted mode.

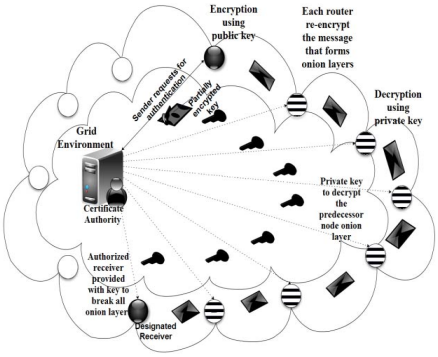

Fig. 1. Onion routing process

(4) After the service provider(s) complete the processing, the response is sent back to the service requestor back in the same path as that established during request sending.

(5) While travelling, the encrypted layer called Onion layer is formed.

(6) At each layer, encryption is done for the messages using ECC algorithm.

(7) The collation may require authenticating the ticket, which is all along flowing with the messages.

(8) The final results are sent back to the edge node.

(9) Edge node in turn sends responses back to the requestor.

(10) At the receiving end, each onion layer peel off and messages will be received.

However in a practical scenario, intermediate nodes may not be available and therefore the encrypted message may get lost. This situation can be better handled by the introduction of Certificate Authority in the onion routing Grid framework, where every time a node gets a message, it registers the sender and receiver information along with the dynamically generated ticket.

Every time the ticket gets changed or the intermediate nodes fail or become unavailable, the node that currently holds the message can query the Certificate Authority to fetch the path of message traversal. By this technique the request receiving path and the response-sending path may not be the same. The encrypted messages while transacting makes sure that each node only knows the next node in the chain with the contents of the messages remaining protected. The dynamic ticket gets ever changing. Based on the request the Certificate Authority provides the authenticated certificate for communication.

IV. Performance Analysis of Ecc and Rsa

ECC not only uses smaller keys for equivalent strength compared to traditional public- key cryptosystems like RSA, the key size disparity grows as security needs increase. This makes it especially attractive for constrained wireless devices because smaller keys result in power, bandwidth and computational savings. More importantly, the advantage of ECC over its competitor's increases as security needs increase over time. Secure Sockets Layer [9] is the most widely deployed and used security protocol on the Internet today. The protocol has withstood years of scrutiny by the security community and is now trusted to secure virtually all sensitive web-based applications ranging from online banking and stock trading to e-commerce. SSL layers encryption, source authentication and integrity protection for data exchanged over insecure, public networks. It operates above a reliable transport service like TCP and has the exibility to accommodate different cryptographic algorithms for key agreement, encryption and hashing. The combinations of algorithms are called as cipher suite.

For example, a cipher suite such as RSA-RC4-SHA would indicate RSA as the key exchange mechanism, RC4 for bulk encryption, and SHA for hashing. In our approach ECC-RIPEMD-ECDSA is used. ECDSA which means ECC Digital Signature Algorithm and RIPEMD is a hash algorithm which helps to convert variable sized messages into fixed sized messages.

TABLE II: Cryptographic Operations is SSL Handshake. Only Sender Authenticated

	RSA	ECDCA
Sender	RSAverify + RSAencrypt	ECDSAverify
Receiver	RSAdecrypt	ECDSAop

(b)Both Sender and Receiver authenticated

	RSA	ECDCA
Sender	RSAverify + RSAencrypt + RSAsign	(i)ECDSAverify or (ii) ECDSAverify + ECDSAsign + ECDSAop
Receiver	2 * RSAverify + RSAdecrypt	(i)ECDSAverify or (ii) 2*ECDSAverify + ECDSAop

A. RSA Handshake

The table 1 shows about the sender performance of two RSA public-key operations one is to verify the receiver's certificate and another to encrypt the premaster secret with the receiver's public key. The receiver only performs one RSA private-key operation to decrypt the sender Key Exchange message and recover the premaster secret.

B. ECDSA Handshake

The sender performs ECDSA verification with receiver's public key to compute the shared premaster. All the receiver needs to do is perform an ECDSA operation to arrive at the same secret.

B. Both Sender and Receiver Authentication

C. RSA Handshake

The sender performs two RSA public-key operations but additionally performs an RSA private-key operation to generate the Certificate Verify message The receiver performs two RSA public-key operations (one to verify the sender's certificate and another to verify the sender's signature) and a private key operation to decrypt the premaster secret.

D. ECDSA Handshake

(i) When the sender uses ECDSA verification operation on the other's certificate followed to compute the premaster secret.

(ii) When the sender uses an ECDSA certificate, the operations required on the two sides are asymmetric. The sender performs an ECDSA verification of the receiver's certificate to compute the premaster secret and an ECDSA signature to generate the Certificate Verify message. The receiver performs two ECDSA verifications one is to verify the sender's certificate and another to verify the Certificate Verify message.

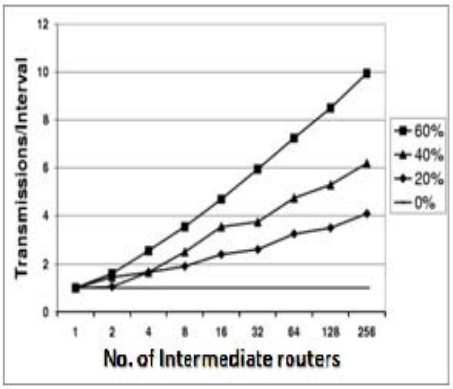

Fig. 2. Performance improvements in ECC.

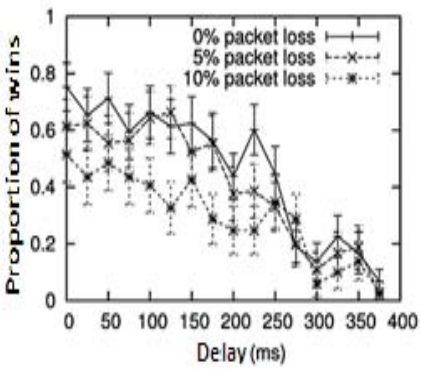

Fig. 3. Comparison of ECC and RSA

V. Conclusions

The above analysis suggests that the use of ECC cipher suites can over significant performance benefits to SSL senders and receivers especially as security needs increase. The concept of Onion Routing has used with enhanced features such as anonymity, unlinkability and inter-nodal encryption and merging the same with the existing features of privacy protection, trust, integrity, confidentiality and authorization in Service Oriented Computational Grid. Figure 2 shows the performance improvements using ECC algorithm through the number of intermediate routers. Figure 3 shows that delay factors between RSA and ECC. The packet loss probability can be reduced using an Onion Routing network during traversal even if the intermediate nodes fail or get malicious. Certificate Authority, where it can register message sender and receiver details and also can get the updated refreshed token periodically. We present the estimation of the performance improvements of onion routing using ECC as well as the comparison graph between performance level of RSA and ECC.

This paper discussed the importance of supporting mobile nodes in VO. The notion of VO is significant in that it could serve as a basic framework for implementing geographically distributed, cross-organizational application systems in a highly flexible manner. To enhance the pervasiveness of VO further, it is of great importance that participation of mobile computing nodes be supported. The extension of the above work using mobile node is to ensure the fail-safe mechanism without compromising privacy and establishing anonymity. While implementing for mobile devices should also consider the possible issues such as reduced CPU performance, small secondary storage, heightened battery consumption sensitivity, and unreliable low-bandwidth communication.

References

[1] Enhancing Grid Security Infrastructure to Support Mobile Computing Nodes-K. Chae and M. Yung (Eds.): WISA 2003, LNCS 2908, *Springer-Verlag Berlin Heidelberg*, pp. 42–54, 2004.2004.

[2] Privacy Protection in Anonymous Computational Grid Services-Debasish Jana Amritava Chaudhuri and Bijan Bihari Bhaumik, TCS 2009.

[3] Performance Analysis of Elliptic Curve Cryptography for SSL- Vipul Gupta, September 28, 2002, Atlanta, Georgia, USA.

[4] Key Sizes Selection in Cryptography and Security Comparison between ECC and RSA- M.J. Wiener, *Efficient DES key search*, manuscript, Bell-Northern Research, August 20, 1993.

[5] Key Sizes Selection in Cryptography and Security Comparison between ECC and RSA- M. J. Wiener, Efficient DES key search, manuscript, Bell-Northern Research, August 20, 1993.

[6] Elliptic Curve Cryptography-An Implementation Tutorial-Anoop MS, *Tata Elxsi Ltd, Thiruvananthapuram*, India, 1997.

[7] Speeding up Secure Web Transactions Using Elliptic Curve Cryptography-Vipul Gupta, Douglas Stebila, Nils Gura, Hans Eberle Sun Microsystems, Inc.

[8] F. Yan, H. Zhang, Z. Shen, L. Zhang, W. Qiang, "An Improved Wireless GSI based on Trusted Computing Technology," *IEEE* 2006.

[9] A. Frier, P. Karlton and P. Kocher, The SSL3.0 Protocol Version 3.0", see http://home.netscape.com /eng/ssl3/.

[10] S. P. Ahua and J. R. Myers, "Survey on wireless Grid Computing," *Journal on Super computing*, Springer Science, 2006.

H. Parveen Begam pursuing her PhD in Information and Communication Engineering and working as associate professor with Computer Science and Engineering at the Anna University. Her research interests include parallel and distributed processing, Mobile Computing, Grid Computing and their applications. Currently, her research focuses on security issues on mobile grid environment. She is a member of the IEEE, ISTE,CSI and ISCA

M. A. Maluk Mohammed received his PhD Degree in Indian Institute of Technology, Chennai at 2006. He is presently working as professor in department of Computer Science and Engineering. His research interests include distributed computing, Grid Computing, Wireless Sensor Netwroks, Mobile Computing and Cluster Computing. He is a member of IEEE, ACM, IACSIT, ISTE, CSI, IARCS