Efficient Transmission of PKI Certificates using Elliptic Curve Cryptography and its Variants

Shivkumar Selvakumaraswamy[1] and Umamaheswari Govindaswamy[2]
[1]Department of ECE, Anna University, India
[2]Department of ECE, PSG College of Technology, India

Abstract: *The demand for wireless networks is increasing rapidly and it becomes essential to design existing Public-Key Infrastructure (PKI) useful for wireless devices. A PKI is a set of procedures needed to create, distribute and revoke digital certificates. PKI is an arrangement that binds public keys with respective user identities by means of a Certificate Authority (CA). The user identity must be unique within each CA domain. The third-party Validation Authority (VA) can provide this information on behalf of CA. The binding is established through the registration and issuance process which is carried out by software at a CA or under human supervision. Elliptic Curve Cryptography (ECC) is proved to be the best suited one for resource constrained applications. This paper compares the two PKI Algorithms ECC and Rivest-Shamir-Adleman (RSA). It is found that ECC-based signatures on a certificate are smaller and faster to create; and the public key that the certificate holds is smaller as well. Verification is also faster using ECC-based certificates, especially at higher key strengths. The security of ECC systems is based on the Elliptic Curve Discrete Logarithm Problem (ECDLP), rather than the integer factorization problem. This allows for faster computations and efficient transmission of certificates.*

Keywords: *ECC, PKI, wireless application protocol, registration authority, digital signature.*

Received September 5, 2013; accepted December 24, 2013; published online March 8, 2015

1. Introduction

A Public Key Infrastructure (PKI) is a system that provides for trusted third party user identity inspection and assurance. Normally, this is done by a Certificate Authority (CA) and uses cryptography involving public and private keys. Digital certificates are electronic credentials that are used to verify the identities of individuals and devices. They have one primary function: To attest that the identification of a user and data is bound to the public key. Like any physical identification, they can be created, expired, renewed, revoked or suspended. Digital certificates are typically issued by recognized organizations that can verify the certificate requestor's identity-a business, a government agency, or a bank. This ensures that once the certificate is issued, the identity of the requestor can be trusted by all who trust the authority. Digital certificates are made up of three parts. The first part of a certificate is the identification information for the user or device. Second is a public key, associated with a private key that is kept by the user. The third part of the digital certificate is the digital signature. Figure 1 shows the process of signature generation and verification.

To date, Rivest-Shamir-Adleman (RSA) and Digital Signature Algorithm (DSA) have enjoyed wide use in applications ranging from anti-cloning to secure firmware updates. However, changing security and performance requirements, as well as the shift to smaller, more mobile devices, are pointing towards a need for smaller and faster signatures. Moreover, the National Security Agency (NSA) requires that signatures used today last for the next 50 years. These requirements pave the way for digital certificates that use Elliptic Curve Cryptography (ECC)-based signatures. ECC and RSA are the two mainly used public key cryptosystems. Each has its own DSA. Elliptic Curve RSA (ECDSA) is the one used by ECC whereas RSA itself can be used for digital signature generation. Another algorithm namely DSA can be used with both RSA and ECC. Using ECC-based signatures with digital certificates provides added size and performance advantages. ECC provides the same level of security as RSA but, for smaller key sizes as shown in Table 1. The rest of the paper is organised as follows. Related works are presented in section 2, section 3 elaborates the basics of ECC. Section 4 discusses ECC based certificates and simulation results are presented in section 5. Conclusions are presented in section 6.

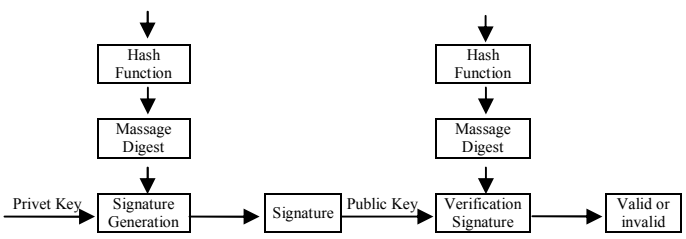

Figure 1. Signature generation and verification.

Table 1. Comparable key sizes (in bits).

Symmetric Cipher	ECC	RSA
80	163	1024
112	233	2240
128	283	3072
192	409	7680
256	571	15360

2. Related Works

Nils *et al*. [9] stated that the relative performance advantage of ECC point multiplication over RSA modular exponentiation increases with the decrease in processor word size and the increase in key size. Elliptic curves over fields using pseudo-mersenne primes standardized by NIST and SECG allow for high performance implementations and show no performance disadvantage over optimal extension fields or prime fields selected specifically for particular processor architecture.

An attempt has been made by Padmavathi and Lavanya [10] to identify a suitable asymmetric-threshold based cryptosystems for small Manets. Different small network scenarios with variable node sizes and key sizes are experimented and the results show that elliptic curve threshold cryptography is the most desirable asymmetric-threshold cryptosystem for small MANET.

ECC is proving to be more secure and many implementations based on ECC is coming up. William and Mohammed [11] details the design of a new high-speed pipelined Application-Specific Instruction set Processor (ASIP) for ECC using Field-Programmable Gate-Array (FPGA) technology. Different levels of pipelining were applied to the data path to explore the resulting performances and find an optimal pipeline depth. Three complex instructions were used to reduce the latency by reducing the overall number of instructions and a new combined algorithm was developed to perform point doubling and point addition using the application specific instructions.

Yuan *et al.* [12] designed and implemented a CA based on ECC by using Java programming technique, which can sign X.509v3 digital certificate to client and then validate client certificate. In application, the key pair (including public key and private key) can be got from the public key cryptography standard (PKCS#12), which is used in encryption, decryption and digital signature. The ECC-based CA is used in the system of Digital Rights Management (DRM) to contribute to confidentiality, authenticity, integrity and non-repudiation in communication.

Mohsen and Ali [8] introduced a Lightweight Public Key Infrastructure (LPKI) for the constrained platforms such as mobile phones. It takes advantages of ECC and signcryption to decrease the computational costs and communication overheads, and adapting to the constraints. LPKI is so suitable for mobile environments and for applications such as mobile commerce where the security is of great concern.

Secure Electronic Transaction (SET) is a standard protocol for the credit card transaction in e-commerce. The public key and private key of cardholder, merchant, payment gateway and CA were distributed based on ECC. Security analysis, done by Cao [2], shows that this scheme has high security and efficient authentication.

The use of X.509v3 certificates to carry out authentication tasks is an approach to improve security. These certificates are usually employed with the RSA algorithm. ECC is well suited for small devices, like those used in wireless communications and is gaining momentum. The study carried out by Maria *et al.* [6] aims to design and implement a free open-source certification authority able to issue X.509v3 certificates using ECC.

3. Elliptic Curve Cryptography

An elliptic curve is not an ellipse, but is represented as a looping line intersecting two axes. ECC is based on properties of a particular type of equation created from the mathematical group derived from points where the line intersects the axes. Multiplying a point on the curve by a number will produce another point on the curve, but it is very difficult to find what number was used, even if the original point and the result is known. Equations based on elliptic curves have a characteristic that is very valuable for cryptography purposes: They are relatively easy to perform and extremely difficult to reverse [4].

ECC is performed over one of two underlying Galois fields: Prime order fields $GF(p)$ or characteristic two fields $GF(2^m)$. Both fields are considered to provide the same level of security [7], but arithmetic in $GF(2^m)$ can be implemented in hardware more efficiently using modulo-2 arithmetic than prime field. An elliptic curve E over the field $GF(2^m)$ is the set of solutions to the equation.

$$y^2 + xy = x^3 + ax^2 + b \tag{1}$$

Where $a, b \epsilon GF(2^m), b \neq 0$.

If $P=(x_1, y_1)$ and $Q=(x_2, y_2)$ are points on the elliptic curve that satisfy Equation 1 and if $P \neq -Q$ and $P \neq Q$ then $R(x_3, y_3)=P+Q$ (Point addition).

Where

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a \tag{2}$$

$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1 \tag{3}$$

$$\lambda = (y_2+y_1)/(x_2+x_1) \tag{4}$$

If $P=Q$ then $R(x_3, y_3)=2P$ (Point double).

Where

$$x_3 = \lambda^2 + \lambda + a \tag{5}$$

$$y_3 = x_1^2 + (\lambda + 1)x_3 \tag{6}$$

$$\lambda = x_1 + y_1 / x_1 \tag{7}$$

The scalar point multiplication over $GF(2^m)$ can be defined as:

$$tP = P+P+P+\ldots+P \ (t \ times) \tag{8}$$

If $P, Q \epsilon A$, the addition $P+Q$ be a point $-R$ (whose inverse is R with only changing the sign of y coordinate

value and lies on the curve) on the E/FP such that all the points P, Q and $-R$ lie on the straight line, i.e., the straight line cuts the curve at P, Q and $-R$ points. If $P=Q$, it becomes a tangent at P or Q, which is assumed to intersect the curve at the point 0. The security strength of the ECC lies on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) and it provides same level of security of RSA with less key size as shown in Table 1.

4. ECC Based Certificates

A CA, which is the base of a PKI, is a trusted agency that issues digital certificates for the verification and validation of user's public keys with the owners of the certificates [3]. The entities individually contact a CA by providing their identity such as name, address, date of birth, public key etc., of each entity. After validation through handshake procedure, the CA performs some level of entity authentication according to Certificate Practices Statement (CPS) and then issues a digital certificate containing the entity's identity and public key for each entity. The entities can now exchange the certificates among themselves and become authenticated to each other. This assures the reception of authenticated public keys of the entities. To minimize the workload of CA, many PKI considers the existence of a Registration Authority (RA) separated from the CA to perform authentication, verification, distribution, revocation, reporting etc., where each RA being certified by CA has both authenticated private and public keys for works together with CA [3].

There are three algorithms that are used for digital certificate generation and verification: RSA, DSA and ECDSA. Other signature algorithms include Elliptic Curve Pintsov Vanstone Signatures (ECPVS) which provides partial message recovery, Elliptic Curve Qu Vanstone (ECQV) which is an implicit certificate type and Elliptic Curve Nyberg Rueppel (ECNR) which provides full message recovery. All the three algorithms like RSA, DSA and ECDSA are utilized for our simulation purpose. They are used in combination with SHA-1, MD5, ECIES and RSA. ECIES and RSA are used for encryption purpose. SHA1 and MD5 are used to produce 'hash' of a message [5].

The *PKI* working group of Internet Engineering Task Force (IETF) specifies ECC based public key certificate standardized by the PKI-X.509 similar to the X.509 RSA based public key certificate [3]. Subsequently, the ECDSA, Elliptic Curve Diffie-Hellman (ECDH) key exchange and the generation of ECC based certificate with ECDSA signature of PKIX are proposed. The ECC-based PKIX is easily interoperable with RSA-based PKI (X.509) and the CA issues and certifies ECC based certificates. A simple ECC-based X.509 certificate format to combine user's identity and the ECC-based public key proposed by International Telecommunication Union (ITU) is shown in Figure 2. The following subsections present ECDSA and RSA Algorithms.

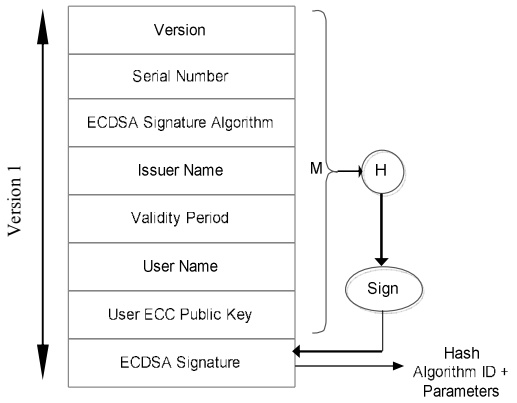

Figure 2. ECC-based X.509 certificate.

4.1. ECDSA Signature Generation and Verification Method

To sign a message m, an entity A with domain parameters $D=(q, FR, a, b, G, n, h)$ does the following:

- Select a random or pseudorandom integer k in the interval $[1, n-1]$.
- Compute $k.G=(x_1, y_1)$ and convert x_1 to an integer x_1.
- Compute $r= x_1 \bmod n$. If $r=0$ then go back to step 1.
- Compute $k^{-1} \bmod n$.
- Compute hash (m) and convert this bit string to an integer.
- Compute $s= k^{-1} \{e+d.r\} \bmod n$. If $s=0$, then go to first step.
- A's signature for the message m is the pair of integers (r, s).

To verify A's signature (r, s) on m, B obtains an authenticated copy of A's domain parameters $D=(q, FR, a, b, G, n, h)$ and associated public key Q and does the following:

- Verify that r and s are integers in the interval [1, n-1].
- Compute hash (m) and convert this bit string into an integer e.
- Compute $w=(s^{-1}) \bmod n$.
- Compute $u_1=e\ w \bmod n$ and $u_2=r\ w \bmod n$.
- Compute $X=u_1G+ u_2G$.
- If $X=0$, then reject the signature, else convert the x coordinate of X to an integer x_1, and compute $v= x_1, \bmod n$.
- Accept the signature if and only if $v=r$.

4.2. RSA Signature Generation and Verification Algorithm

RSA is one of the oldest and most widely used public key cryptographic Algorithms. The algorithm was invented in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman. The RSA cryptosystem is based on the assumption that factoring is a computationally hard task. This means that given sufficient computational resources and time, an adversary should not be able to "break" RSA (obtain a private key) by factoring. This does not mean that factoring is the only way to "break" RSA. In fact, breaking RSA may be easier than factoring.

RSA public and private key pair can be generated using the algorithm given below:

- Choose two random prime numbers p and q.
- Compute n such that $n=p*q$.
- Compute $\varphi(n)$ such that $\varphi(n)=(p-1)*(q-1)$.
- Choose a random integer e such that $1< e<\varphi(n)$ and $gcd(e, \varphi(n))=1$, then compute the integer d such that: $e*d=1 \bmod \varphi(n)$.
- (e, n) is the public key and (d, n) is the private key.

Signature of a message 'm' is a straightforward modular exponentiation using the hash of the message and the private key. The signature s can be obtained by:

$$s = hash(m)^{d} \bmod n \quad (9)$$

A common hash algorithm used is SHA-1. To verify a signature s for message m, the signature must first be decrypted using the author's public key (e, n). The hash h is thus obtained by:

$$h = s^{e} \bmod n \quad (10)$$

If h matches hash (m), then the signature is valid. The message was signed by the author and the message has not been modified since signing.

5. Simulation Results and Discussion

Network Simulator NS2 is used for creating wireless network environment and simulations. Both the physical layer and the 802.11 MAC layer are included in the wireless extension of NS2. The moving speed of mobile node is limited to 20m/s and a pause time of 1000 s. User Datagram Protocol traffic with constant bit rate is implemented with a packet size of 512bytes. Total no of nodes is 50. Random waypoint model is used for simulation.

For key generation ECC and RSA are used. SHA1 is the commonly used algorithm for message digest generation and it is compared with MD5. These two Algorithms are used with ECC and RSA and its performance is compared. For encryption and decryption we have chosen ECIES, which is an elliptic curve based algorithm and RSA. ECDSA and RSA are used for digital signature generation. The time involved in generating the certificate for ECC based Algorithms and RSA based algorithm is measured. It is found that the ECC based schemes take lesser time for certificate generation compared to RSA algorithm. The measured time for certificate generation using RSA and ECC Algorithms for comparable key sizes are shown in Table 2.

Table 2. Certificate generation timings.

Length of the Key(Bits)		Certificate Generation	
ECC	RSA	RSA Time (ms)	ECC Time (ms)
163	1024	1145	215
233	2340	1215	238
283	3072	1350	250
409	7680	1430	270

A CA server generates certificates upon the request from the client. The time elapsed between the client requesting for the certificate and the server issuing the certificate to the client is measured as the transmission of certificates between the client and server. The time measurements are given in Table 3. There is a huge difference in the time consumption between RSA and ECC Algorithms. This is due to the exponentiation present in the RSA algorithm. This takes a longer computation time. Also, the certificate generated by ECC holds smaller keys compared to RSA.

Table 3. Transmission of certificates to clients.

Length of the Key(Bits)		Transmission of Certificates	
ECC	RSA	RSA Time(s)	ECC Time(s)
163	1024	102	0.166
233	2340	105	0.332
283	3072	112	0.381
409	7680	112	0.45

The time taken for key generation and encryption for RSA and ECC Algorithms are given in Tables 4 and 5, respectively. The result produced by our article is compared with the results produced by Alese *et al.* [1]. Our implementation has produced lesser time compared to [1] due to code compactness and optimized code simulation present in our method.

Table 4. Key generation and encryption time for RSA.

Key Size (Bits)	Alese *et al.* [1]		This Paper	
	Key Generation (ms)	Encryption (ms)	Key Generation (ms)	Encryption (ms)
1024	1312.7	166.9	1274	146
2048	6804.6	290.2	6234.3	220.3
3072	32108	310.5	28396	265.3
7680	322843	352.1	289232	420

Table 5. Key generation and encryption time for ECC.

Elliptic Curve	Alese *et al.* [1]		This Paper	
	Key Generation (ms)	Encryption (ms)	Key Generation (ms)	Encryption (ms)
P-160	198.6	15.7	125.3	13.2
P-224	208.3	18.8	131	22
P-256	243.5	25	234.4	26
P-384	294.0	50.1	276.7	42

The computational complexity increases with increase in key size as shown in Figure 3. This is only an indicative figure which shows that larger key sizes require more computation time. With the presence of exponent calculation RSA takes more time compared to ECC (not shown in figure). Figure 4 gives the comparison of time taken for certificate generation by RSA and ECC algorithm for a given key size. The time taken for the transmission of certificate from sender to receiver is shown in Figure 5. ECC takes lesser time for various key sizes and is much faster than RSA.

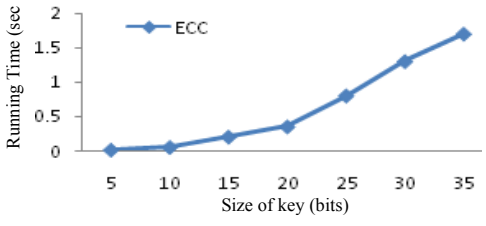

Figure 3. Running time versus key size.

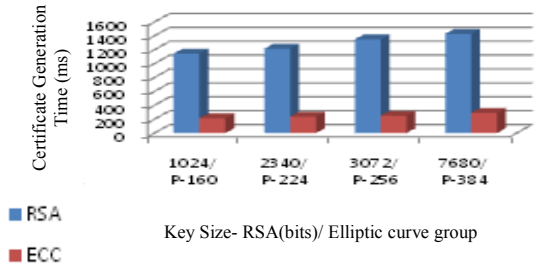

Figure 4. Certificate generation time-comparison of ECC and RSA Algorithms.

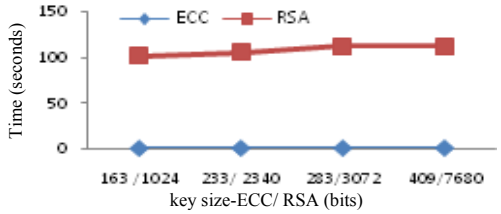

Figure 5. Transmission of certificate to clients.

Public key cryptosystem utilizes two different keys: Private key and public key. RSA and ECC have their own key generation Algorithms. We have compared the time taken by RSA and ECC Algorithms implemented by us with the results of [1]. The RSA key generation time is shown in Figure 6. Our implementation is faster for the given RSA key sizes. The encryption time is compared in Figure 7. Similarly, the key generation time and encryption time for ECC is compared in Figures 8 and 9 respectively. Faster key generation and encryption is due to compact and optimized code simulation present in our method.

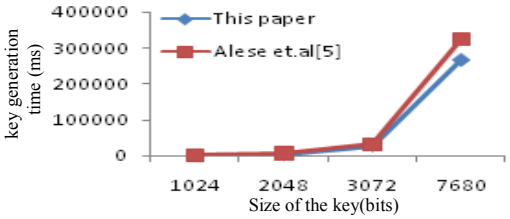

Figure 6. Key generation time for RSA.

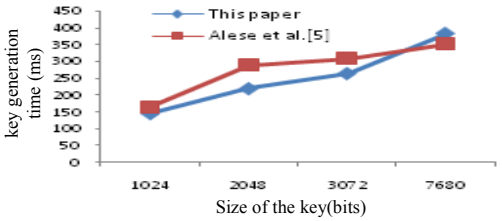

Figure 7. Encryption time for RSA.

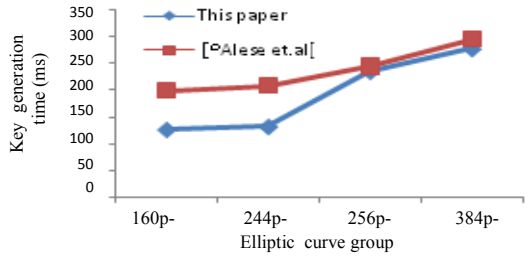

Figure 8. Key generation time for ECC.

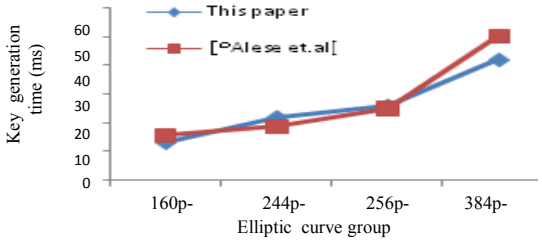

Figure 9. Encryption time for ECC.

6. Conclusions

The ECC-based signatures are smaller and faster to create than RSA-based Algorithms. The public key that the certificate holds is smaller and more agile as well. Verification is also faster using ECC based certificates, especially at higher key strengths. In addition, public- and private-key operations impose different processor loads: For RSA, public key operations impose a lower load than private key ones. For ECC, private key operations impose a slightly lower load than public key ones. The combination of Elliptic curve and SHA-1 algorithm provides strong cryptographic strength and optimizes the computational speed as well as space. Our results are compared with other results. We are able to do key generation and encryption much faster than other implementations. This is due to compact and optimized code simulation. As the ECC is based on the strength of the ECDLP, it is not vulnerable to cryptanalysis attacks which are readily available. This makes ECC based schemes more suited for time and resource constrained wireless applications.

References

[1] Alese K., Philemon E., and Falaki S., "Comparative Analysis of Public-key Encryption Schemes *International Journal of Engineering and Technology*, vol. 2, no. 9, pp. 1552-1568, 2012.

[2] Cao L., "Improving Security of SET Protocol based on ECC," *in Proceedings of the International Conference on Web Information Systems and Mining*, Taiyuan, pp. 234-241, 2011.

[3] Dasd C., Farrell S., Kause T., and Mononen T., "Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP): IETF RFC 2510," available at: https://tools.ietf.org/html/rfc4210, last visited 2005.

[4] Hankerson D., Menezes A., and Vanstone S., *Guide to Elliptic Curve Cryptography*, Springer-Verlag, New York, USA, 2004.

[5] Lakshmanan T. and Muthusamy M., "A Novel Secure Hash Algorithm for Public Key Digital Signature Schemes," *the International Arab Journal of Information Technology*, vol. 9, no. 3, pp. 262-267, 2012.

[6] Maria C., Ruben V., and Fernando C., "A Certification Authority for Elliptic Curve X.509v3 Certificates," *in Proceedings of the 3rd*

International Conference on Networking and Services, Athens, pp. 49-49, 2007.

[7] Miller S., "Use of Elliptic Curves in Cryptography," *Advances in Cryptology-CRYPTO '85*, Springer Berlin Heidelberg, 1985.

[8] Mohsen T. and Ali B., "LPKI-A Lightweight Public Key Infrastructure for the Mobile Environments," *in Proceedings of the 11th IEEE International Conference on Communication Systems*, Guangzhou, pp. 162-166, 2008.

[9] Nils G., Arun P., Arvinderpal W., Hans E., and Sheueling S., "Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs," *in Proceedings of the 6th International Workshop Cambridge*, USA, pp. 119-132, 2004.

[10] Padmavathi G. and Lavanya B., "Comparison of RSA Threshold Cryptography and ECC Threshold Cryptography for Small Mobile Ad-hoc Networks," *International Journal of Advanced Networking and Applications*, vol. 3, no. 4, pp. 1245-1252, 2012.

[11] William C. and Mohammed B., "Fast Elliptic Curve Cryptography on FPGA," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 16, no. 2, pp. 198-205, 2008.

[12] Yuan Y., Liu Q., and Li F., "A Design of Certificate Authority Based on Elliptic Curve Cryptography," *in Proceedings of the 9th International Symposium on Distributed Computing and Applications to Business Engineering and Science*, Hong Kong, pp. 454-457, 2010.

Shivkumar Selvakumaraswamy is a PhD scholar of Anna University, Chennai, India. He received his BE degree in electronics and communication engineering from Bharathiar University, India, in 1995 and ME degree in communication systems from Anna University, India in 2005. He has teaching experience of more than 18 years and a member of ISTE. His area of interest includes wireless networks, information security and communication networks.

Umamaheswari Govindaswamy received her BE degree in electronics and communication engineering from Madras University, India, in 1989 and ME degree in electronics engineering from Anna University, India, in 1992 and PhD from Bharathiar University, India, in 2006. She has teaching experience of more than 20 years. Her area of interest includes image processing, data communication and information security. She has published 40 papers in National and International Conferences and Journals.