Providing Information Security Using Public Key Cryptosystems

Mr. D.Shiva Rama Krishna, Mr. Siva Rama Prasad Kollu , Mr. Ch.V.V.Narasimha Raju

Abstract— Information security is main issue of this generation of computing because many types of attacks such as passive and active are increasing day by day. Information security is the practice of defending information from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction, denial of message. It is a general term that can be used regardless of the form the data may providing confidentiality, integrity, and availability of information is is at the heart of information security to provide all these things in this paper we proposed public key cryptosystems. Public-key cryptography, also known as asymmetric cryptography, is a class of cryptographic protocols based on algorithms that require two separate keys , one of which is secret (or private) and one of which is public. To provide confidentiality, integrity, and availability of information we are using public key cryptosystems such as RSA algorithm, Elliptic curve cryptography, Diffie-Hellman key exchange algorithm and Digital Signatures.

Index Terms— RSA, Diffie-Hellman, Digital Signatures, Elliptic curve cryptography

I. INTRODUCTION

Now a day's providing security to confidential information is necessary and crucial thing. Information security,, is the practice of defending information from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction. It is a general term that can be used regardless of the form the data may take (e.g. electronic, physical).to provide the security we are using public key cryptosystems. Public-key cryptography, also known as asymmetric cryptography, is a class of cryptographic protocols based on algorithms that require two separate keys, one of which is secret (or private) and one of which is public. Although different, the two parts of this key pair are mathematically linked. The public key is used, for example, to encrypt plaintext or to verify a digital signature; whereas the private key is used for the opposite operation, in these examples to decrypt cipher text or to create a digital signature. The term "asymmetric" stems from the use of different keys to perform these opposite functions, each the inverse of the other – as contrasted with conventional ("symmetric") cryptography which relies on the same key to perform both. Public-key algorithms are based on mathematical problems that currently admit no efficient solution and are inherent in certain integer factorization, discrete logarithm, and elliptic curve relationships. It is computationally easy for a user to generate their own public and private key-pair and to use them for encryption and decryption. The strength lies in it being "impossible" (computationally infeasible) for a properly generated private key to be determined from its corresponding public key. Thus the public key may be published without compromising security, whereas the private key must not be revealed to anyone not authorized to read messages or perform digital signatures. Public key algorithms, unlike symmetric key algorithms, do not require a secure initial exchange of one (or more) secret keys between the parties.

Manuscript received May, 2015.

Mr D.Shiva rama krishna,Computer Science and Engineering, MLR Institute of Technology & management, Hyderabad, India, 9492673201., (e-mail: shivaramakrishna.devalla@mlritm.ac.in).

Mr K.Siva Rama Prasad,Computer Science and Engineering, MLR Institute of Technology & management, Hyderabad, India, 9866213002., (e-mail: siva.rama1350@mlritm.ac.in).

Mr Ch.v.v.Narasimha Raju,Computer Science and Engineering, MLR Institute of Technology & management, Hyderabad, India, 9959969165., (e-mail: chinta88narsi@mlritm.ac.in).

II. APPLICATIONS OF PUBLIC KEY CRYPTOSYSTEMS

Before proceeding, we need to clarify one aspect of public-key cryptosystems that is otherwise likely to lead to confusion. Public-key systems are characterized by the use of a cryptographic algorithm with two keys, one held private and one available publicly. Depending on the application, the sender uses either the sender's private key or the receiver's public key, or both, to perform some type of cryptographic function. In broad terms, we can classify the use of public-key cryptosystems into three categories

• Encryption /decryption: The sender encrypts a message with the recipient's public key.

• Digital signature: The sender "signs" a message with its private key. Signing is achieved by a cryptographic algorithm applied to the message or to a small block of data that is a function of the message.

• Key exchange: Two sides cooperate to exchange a session key. Several different approaches are possible, involving the private key(s) of one or both parties.

Some algorithms are suitable for all three applications, whereas others can be used only for one or two of these applications. Table 1 indicates the applications supported by the algorithms

Algorithm	**Encryption/Decryption**	**Digital Signature**	**Key Exchange**
RSA	Yes	Yes	Yes
Elliptic curve	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No

.

Table 1. Applications for Public-Key Cryptosystems

III. RSA ALGORITHM

One of the first successful responses to the challenge was developed in 1977 by Ron Rivest, Adi Shamir, and Len Adelman at MIT and first published in 1978[RIVE78].5 The Rivest-Shamir-Adelman (RSA) scheme has since that time reigned
supreme as the most widely accepted and implemented general-purpose approach to public-key encryption.
The **RSA** scheme is a block cipher in which the plaintext and cipher text are integers between 0 and n - 1 for some n.A typical size for n is 1024 bits, or 309 decimal digits. That is, n is less than 2^{1024}.We examine RSA in this section in some detail, beginning with an explanation of the algorithm. Then we examine some of The computational and crypt analytical implications of RSA.

Description of the Algorithm
RSA makes use of an expression with exponentials. Plaintext is encrypted in blocks, with each block having a binary value less than some number n.That is, the block size must be less than or equal to $\log_2(n) + 1$; in practice, the block size is i bits, where $2i < n \leq 2i+1$. Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C.

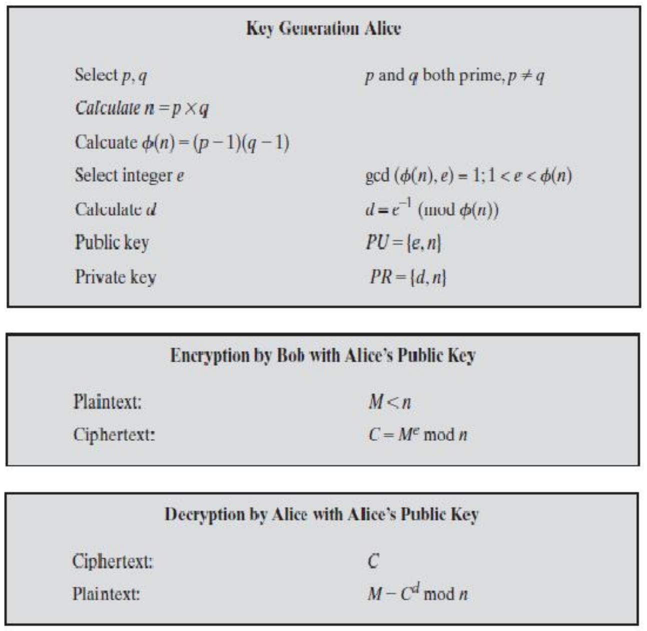
$C = M^{e} \bmod n$
$M = C^{d} \bmod n = (M^{e})^{d} \bmod n = M^{ed} \bmod n$

Both sender and receiver must know the value of n. The sender knows the value of e, and only the receiver knows the value of d. Thus, this is a public-key encryption algorithm with a public key of $PU = \{e, n\}$ and a private key of $PR = \{d, n\}$.
For this algorithm to be satisfactory for public-key encryption, the following requirements must be met.
1. It is possible to find values of e, d, n such that $M^{ed} \bmod n = M$ for all $M < n$.
2. It is relatively easy to calculate $M^{e} \bmod n$ and $C^{d} \bmod n$ for all values of $M < n$.
3. It is infeasible to determine d given e and n.
For now, we focus on the first requirement and consider the other questions later. We need to find a relationship of the form
$M^{ed} \bmod n = M$
The preceding relationship holds if e and d are multiplicative inverses modulo $\Phi(n)$, where $\Phi(n)$ is the Euler totient function. for p, q prime, $\Phi(pq) = (p - 1)(q - 1)$. The relationship between e and d can be expressed as
$ed \bmod \Phi(n) = 1$
This is equivalent to saying
$ed \equiv 1 \bmod \Phi(n)$
$d \equiv e^{-1} \bmod \Phi(n)$
That is, e and d are multiplicative inverses mod $\Phi(n)$. Note that, according to the rules of modular arithmetic, this is true only if d (and therefore e) is relatively prime to $\Phi(n)$. Equivalently, gcd $(\Phi(n), d) = 1$.We are now ready to state the RSA scheme. The ingredients are the following:

p,q are two prime numbers	(private, chosen)
n=pq	(public, calculated)
e, with $\gcd(\Phi(n),e)=1; 1<e<\Phi(n)$	(public, chosen)
$d \equiv e^{-1} \bmod \Phi(n)$	(private, caluclated)

The private key consists of $\{d, n\}$ and the public key consists of $\{e, n\}$. Suppose that user A has published its public key and that user B wishes to send the message M to A. Then B calculates $C = M^{e} \bmod n$ and transmits C. On receipt of this cipher text, user A decrypts by calculating $M = C^{d} \bmod n$.
Figure 1 summarizes the RSA algorithm. It corresponds to Figure1 Alice Generates a public/private key pair; Bob encrypts using Alice's public key; and Alice decrypts using her private key. An example from [SING99] is shown in Figure 2 For this example, the keys were generated as follows:
1. Select two prime numbers, $p = 17$ and $q = 11$.
2. Calculate $n = pq = 17 \text{ X} 11 = 187$.
3. Calculate $\Phi(n) = (p - 1)(q - 1) = 16 \text{ X} 10 = 160$.
4. Select e such that e is relatively prime to $\Phi(n)= 160$ and less than $\Phi(n)$; we Choose $e = 7$.
5. Determine d such that $de \equiv 1 \pmod{160}$ and $d < 160$.The correct value is $d = 23$, Because 23 X 7 = 161 = (1X160) + 1; d can be calculated using the extended Euclid's algorithm.

The resulting keys are public key $PU = \{7, 187\}$ and private key $PR = \{23, 187\}$.
The example shows the use of these keys for a plaintext input of M= 88. For encryption, we need to calculate C = 887 mod 187. Exploiting the properties of modular Arithmetic, we can do this as follows:
$88^{7} \bmod 187 = [(88^{4} \bmod 187) \text{ X}\square(88^{2} \bmod 187) \text{ X } (88^{1} \bmod 187)] \bmod 187$
$88^{1} \bmod 187 = 88$
$88^{2} \bmod 187 = 7744 \bmod 187 = 77$
$88^{4} \bmod 187 = 59{,}969{,}536 \bmod 187 = 132$
$88^{7} \bmod 187 = (88\text{X}77\text{X}132) \bmod 187 = 894{,}432 \bmod 187 = 11$



Key Generation Alice

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calcuate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Encryption by Bob with Alice's Public Key

Plaintext:	$M < n$
Ciphertext:	$C = M^{e} \bmod n$

Decryption by Alice with Alice's Public Key

Ciphertext:	C
Plaintext:	$M - C^{d} \bmod n$

Fig 1: The RSA Algorithm

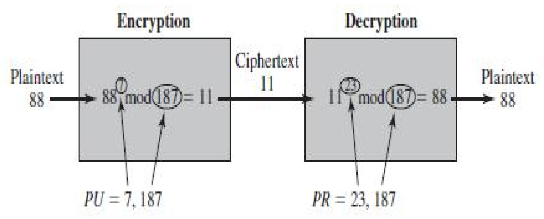


Fig 2: Example of RSA Algorithm

IV. DIFFIE-HELLMEN KEY EXCHANGE ALGORITHM

A simple public-key algorithm is Diffie-Hellman key exchange. This protocol enables two users to establish a secret key using a public-key scheme based on discrete logarithms. The protocol is secure only if the authenticity of the two participants can be established. The scheme was first published by Whitfield Diffie and Martin Hellman in 1976. By 1975, James H. Ellis, Clifford Cocks and Malcolm J. Williamson within GCHQ, the British signals intelligence agency, had also shown how public-key cryptography could be achieved; however, their work was kept secret until 1997.

The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages. The algorithm itself is

limited to the exchange of secret values. The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of

computing discrete logarithms. Briefly, we can define the discrete logarithm in the following way. Recall from Chapter 8 that a primitive root of a prime number as one whose powers modulo generate all the integers from 1 to p-1.That is, if a is a primitive root of the prime number , then the numbers

a mod p, a^2 mod p,………. a^{p-1} mod p.

Are distinct and consist of the integers from 1 through p-1 in some permutation. For any integer b and a primitive root of prime number , we can find a unique exponent such that

$b = a^i \pmod{p}$ where $0 \leq i \leq (p - 1)$.

The exponent is referred to as the discrete logarithm of for the base , mod .We express this value as d log $_{ap}$(b)

For this Diffie-Hellman key exchange algorithm, there are two publicly known numbers: a prime number q and an integer α that is a primitive root of q. Suppose the users A and B wish to exchange a key. User A selects a random integer X_A <q and computes $Y_A= \alpha^{X_A}$ mod q. Similarly, user B independently selects a random integer X_B<q and computes $Y_B= \alpha^{X_B}$ mod q.Each side keeps the X value private and makes the Y value available publicly to the other side. User A computes the key as $K=(Y_B)^{X_A}$ mod q and user B computes the key as $K=(Y_A)^{X_B}$ mod q.These two calculations produce identical results. Figure 3,4 summarizes following calculations:

$$\begin{aligned} K &= (Y_B)^{X_A} \bmod q \\ &= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\ &= (\alpha^{X_B})^{X_A} \bmod q \quad \text{by the rules of arithmetic} \\ &= \alpha^{X_B X_A} \bmod q \\ &= (\alpha^{X_A})^{X_B} \bmod q \\ &= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\ &= (Y_A)^{X_B} \bmod q. \end{aligned}$$

The result is that the two sides have exchanged a secret value. Furthermore,because X_A and X_B are private, an adversary only has the following ingredients to work with: q, α ,Y_A , and Y_B .Thus, the adversary is forced to take a discrete logarithm to determine the key. For example, to determine the private key of user B, an adversary must compute

$$X_B=\text{dlog}_{\alpha,q}(Y_B)$$

The adversary can then calculate the key *K* in the same manner as user B calculates it.

The security of the Diffie-Hellman key exchange lies in the fact that, while it is relatively easy to calculate exponentials modulo a prime, it is very difficult to calculate discrete logarithms. For large primes, the latter task is considered infeasible.

Here is an example. Key exchange is based on the use of the prime number q=353 and a primitive root of 353, in this case α=3.A and B select secret keys , X_A =97 and X_A =233,respectively. Each computes its public key:

A computes $Y_A=3^{97}$mod 353 = 40.

B computes $Y_B=3^{233}$mod 353 =248.

After they exchange public keys, each can compute the common secret key:

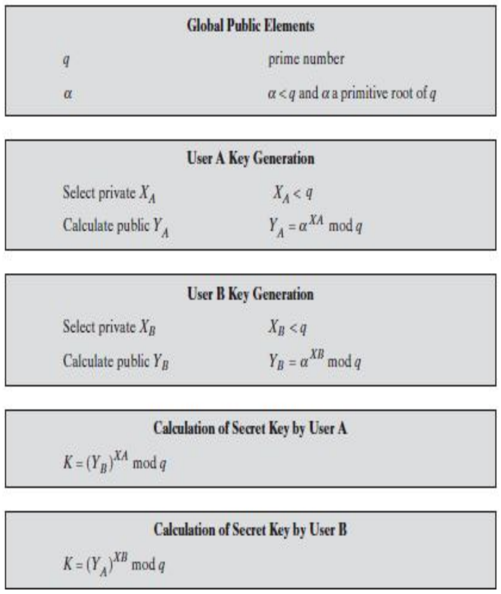
A computes $K=(Y_B)^{X_A}$ mod 353 =248^{97} mod 353=160.

B computes $K=(Y_A)^{X_B}$ mod 353 =40^{233} mod 353=160

We assume an attacker would have available the following information:

q = 353; α = 3; *YA* = 40; *YB* = 248

In this simple example, it would be possible by brute force to determine the secret key 160. In particular, an attacker E can determine the common key by discovering a solution to the equation or the equation 3^a mod 353=40 or the equation 3^b mod 353=248 The brute-force approach is to calculate powers of 3 modulo 353, stopping when the result equals either 40 or 248. The desired answer is reached with the exponent value of 97, which provides 3^{97} mod 353=40. With larger numbers, the problem becomes impractical



Global Public Elements

q	prime number
α	$\alpha < q$ and α a primitive root of q

User A Key Generation

Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{XA} \bmod q$

User B Key Generation

Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{XB} \bmod q$

Calculation of Secret Key by User A

$K = (Y_B)^{XA} \bmod q$

Calculation of Secret Key by User B

$K = (Y_A)^{XB} \bmod q$

Fig 3:Diffe-Hellman Key Exchange Algorithm

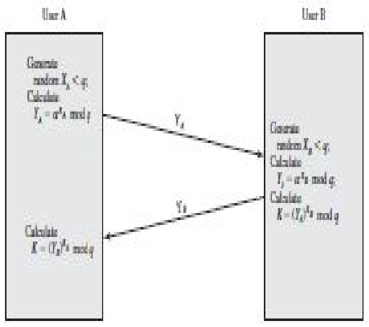

User A
User B
Generate random $X_A < q$;
Calculate $Y_A = \alpha^{X_A} \bmod q$
Y_A
Generate random $X_B < q$;
Calculate $Y_B = \alpha^{X_B} \bmod q$;
Calculate $K = (Y_A)^{X_B} \bmod q$
Y_B
Calculate $K = (Y_B)^{X_A} \bmod q$

Fig 4: Diffe-Hellman Key Exchange

V. DIGITAL SIGNATURE STANDARD

The National Institute of Standards and Technology (NIST) has published Federal Information Processing Standard FIPS 186, known as the Digital Signature Standard (DSS). This Standard specifies a suite of algorithms that can be used to generate a digital signature. Digital signatures are used to detect unauthorized modifications to data and to authenticate the identity of the signatory. In addition, the recipient of signed data can use a digital signature as evidence in demonstrating to a third party that the signature was, in fact, generated by the claimed signatory. This is known as non-repudiation, since the signatory cannot easily repudiate the signature at a later time.

In digital signature standard we are implementing digital signature algorithm. The Digital Signature Algorithm (DSA) is a Federal Information Processing Standard for digital signatures. It was proposed by the National Institute of Standards and Technology (NIST) in August 1991 for use in their Digital Signature Standard (DSS) and adopted as FIPS 186 in 1993.

The Digital Signature Algorithm (DSA) can be used by the recipient of a message to verify that the message has not been altered during transit as well as ascertain the originator's identity. A digital signature is an electronic version of a written signature in that the digital signature can be used in proving to the recipient or a third party that the message was, in fact, signed by the originator. Digital signatures may also be generated for stored data and programs so that the integrity of the data and programs may be verified at any later time.

Digital Signature Generation and Verification

The DSA is used by a signatory to generate a digital signature on data and by a verifier to verify the authenticity of the signature. Each signatory has a public and private key. The private key is used in the signature generation process and the public key is used in the signature verification process. For both signature generation and verification, the data (which is referred to as a message) is reduced by means of the Secure Hash Algorithm (SHA) specified in FIPS 180-1. An adversary, who does not know the private key of the signatory, cannot generate the correct signature of the signatory. In other words, signatures cannot be forged. However, by using the signatory's public key, anyone can verify a correctly signed message.

VI. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. One of the main benefits in comparison with non-ECC cryptography (with plain Galois fields as a basis) is the same level of security provided by keys of smaller size.

Elliptic curves are applicable for encryption, pseudo-random generators and other tasks. They are also used in several integer factorization algorithms that have applications in cryptography, such as Lenstra elliptic curve factorization.

ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers. The technology can be used in conjunction with most public key encryption methods, such as RSA, and Diffie-Hellman. According to some researchers, ECC can yield a level of security with a 164-bit key that other systems require a 1,024-bit key to achieve. Because ECC helps to establish equivalent security with lower computing power and battery resource usage, it is becoming widely used for mobile applications. Elliptic curves are applicable for encryption, digital signatures, pseudo-random generators and other tasks. They are also used in several integer factorization algorithms that have applications in cryptography, such as Lenstra elliptic curve factorization.

VII. CONCLUSION & FUTURE WORKS

From all mentioned public key cryptosystems we can conclude that RSA algorithm can be used for encrypting and signing data. The encryption and signing processes are performed through a series of modular multiplications and also used for providing confidentiality for the information. Diffie-Hellman algorithm is not for encryption or decryption but it enable two parties who are involved in communication to generate a shared secret key for exchanging information confidentially Elliptic Curve Cryptography (ECC) provides similar functionality to RSA. Elliptic Curve Cryptography (ECC) is being implemented in smaller devices like cell phones. It requires less computing power compared with RSA. ECC encryption systems are based on the idea of using points on a curve to define the public/private key pair.\digital Signature Standard (DSS) specifies a suite of algorithms that can be used to generate a digital signature. Digital signatures are used to detect unauthorized modifications to data and to authenticate the identity of the signatory. In addition, the recipient of signed data can use a digital signature as evidence in demonstrating to a third party that the signature was, in fact, generated by the claimed signatory. This is known as non-repudiation, since the signatory cannot easily repudiate the signature at a later time.

In the near future, we can enhance security of confidential information by increased size of keys that are used to perform encryption and decryption of the given information.

VIII. REFERENCES

[1] V.Kavitha & K.S Easwarakumar, (2008) **"**Enhancing Privacy in Arithmetic Coding" ICGSTAIML journal, Volume 8, Issue I.

[2] J.A Storer, (1988) "Data Compression: Methods and Theory" Computer Science Press. Dr. V.K. Govindan & B.S. Shajee mohan "An intelligent text data encryption and compression for high speed and secure data transmission over internet"

[3] Di_e, W., and Hellman, M. New directions in cryptography. IEEE Trans. Inform. Theory IT-22, (Nov. 1976), 644-654.

[4] Di_e, W., and Hellman, M. Exhaustive cryptanalysis of the NBS data encryption standard. Computer 10 (June 1977), 74-84.

[5] D. Djenouri, L. Khelladi, "A Survey of Security Issues in Mobile Ad Hoc and Sensor Networks," IEEE Communication Surveys and Tutorials, vol. 7, no. 4, pp. 2–28, December 2005.

[6] G. Gaubatz, J-P. Kaps, B. Sunar, "Public Key Cryptography in Sensor Networks Revisited", 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004), Lecture Notes in Computer Science, vol. 3313, Springer, Heidelberg, pp. 2-18, August, 2004.

[7] A. Perrig, R. Szewczyk, V. Wen, D. Culler, J. D. Tygar, "SPINS: R. Lehtinen, (2006), "Computer Security Basics", 2nd Edition, O'Reilly, ISBN-10: 0-596-00669-1.

[8]Shobhalokhande,Dipalisawant,NazneenSayyad,MamataYengul," E-Voting through Biometrics and Cryptography-Steganography Technique with conjunction of GSM Modem", Emerging Trends in Computer Science and Information Technology -2012(ETCSIT2012)Proceedings published in International Journal of Computer Applications® (IJCA).

[9] William Stallings,"Cryptography and Network Security, Principles and Practices", Third Edition, pp. 67-68 and 317-375, Prentice Hall, 2003.

[10] T. Kohno, A. Stubblefield, A. D. Rubin, and D. S. Wallach, "Analysis of an Electronic Voting System," Security and Privacy, IEEE Symposium on, vol. 0, p. 27, 2004.

[11] Crypttography and Nettwork SecuriittyFiifftth Ediittiion by Wiilllliiam Sttalllliings Lectture slliides by Lawriie Brown

[12] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, "Handbook of Applied Cryptography", CRC Press, Boca Raton, 1997.

[13] B. A. Forouzan, "Cryptography and Network Security", McGraw Hill, Boston, 2008.

Mr. D.Shiva Rama Krishna, has Bachelors and Master's Degree in the field of Computer science and Engineering. He has keen interest in the area of Cryptography & Network Security, Data Mining, Cloud Computing and has published several papers in National and International conferences and journals. He has attended several workshops and faculty development program.

Mr. Siva Rama Prasad Kollu, has Bachelors in the field of Information Technology and Master's Degree in the field of System Analysis and Computer Applications. He has keen interest in the area of Cryptography & Network Security, Cloud Computing and has published several papers in National and International conferences and journals. He has attended several workshops and faculty development program.

Mr. Ch.V.V.Narasimha Raju, , has Bachelors in the field of Information Technology and Master's Degree in the field of Computer science and Engineering.He has keen interest in the area of Cryptography & Network Security, Data Mining, and has published several papers in National and International conferences and journals. He has attended several workshops and faculty development program.