Performance Evaluation of Modified Signcryption Scheme

Bharat Gupta
Research Scholar, Department of CSE,
Mewar University, Chittorgarh,
Rajasthan, India

Dr. Rajeev Gupta
Professor & Head, Department of ECE,
University College of Engineering,
Rajasthan Technical University, Kota
Rajasthan, India

Abstract – **Before a message is sent out, the sender of the message would sign it using a digital signature scheme and then encrypt the message (and the signature) use a private key encryption algorithm under a randomly chosen message encryption key. The random message encryption key would then be encrypted using the recipient's public key. We call this process two-step approach: "signature-then-encryption". Concept signcryption, first proposed by Zheng, is a cryptography primitive which combines both the functions of digital signature and public key encryption in a logical single step, and with a computational cost and communication overhead are significantly lower than that needed by traditional signature then encryption. In Zheng's scheme, the signature verification can be done recipient's private key at receiver end and its security are based on mainly Discrete Logarithm Problem (DLP), reversing one way hash function. In proposed modified signcryption scheme, security is based on the intractability of three hard problems: Discrete Logarithm Problem, reversing one way hash function and to determine prime factors of a composit number. Proposed signcryption scheme has all the benefits of signcryption and is also able to resolve the dispute / problem of non repudiation by independent third party, without compromising with sender and recipient's private keys.**

Keywords: RSA, ElGamal, Discrete logarithm problem (DLP) ,Cryptography, Secure communication, Digital Signature, Hash, Encryption, Decryption , Signcryption**.**

I. INTRODUCTION:

To avoid forgery and ensure confidentiality to the contents of a letter, for centuries it has been a common practice for the originator of the letter to "sign" his or her name on it and then seal it in an envelope, before handing it over to a deliverer. Then a two-step process "public key cryptography" discovered nearly three decades ago which has revolutionized the way for people to conduct secure and authenticated communications. It became possible for people who have never met before to communicate with one another in a secure and authenticated way over an open and insecure network such as Internet. In doing so, the same two-step approach has been followed.

Signature generation and encryption consume machine cycles and also introduce "expanded" bits to an original message. A comparable amount of computation time is generally required for signature verification and decryption. Hence the cost of a cryptographic operation on a message is typically measured in the message expansion rate and the computational time invested by both the sender and the recipient. With the standard signature-then-encryption approach, the cost for delivering a message in a secure and authenticated way is essentially the sum of the cost for digital signature and for encryption [1-3].

It is possible to transfer a message of arbitrary length in a secure and authenticated way with an expense less than that required by signature-then-encryption. A new cryptographic primitive is termed as "**signcryption**" which simultaneously fulfills both the functions of digital signature and public key encryption in a logically single step, and with a cost significantly smaller than that required by signature-then-encryption [7].

Signcryption techniques generally has a signcrypting algorithm S at the sender end and a unsigncrypting U algorithm at the receiver end has following characteristics :

1.Unique unsigncryptability --Given a message **m** of arbitrary length, the algorithm **S** signcrypts **m** and outputs a signcrypted text **C**. On input **C**, the algorithm **U** unsigncrypts **C** and recovers the original message at the receiver end.

2.Security – At the same time (S, U) fulfills both the properties of a secure encryption scheme and those of a secure digital signature scheme. Attackers cannot find out the message until the private key is known to them and the receiver is sure about whatever is message he / she is getting as a result of U is unforged and signed by an authentic person.

3.Efficiency – Signcryption is economical in terms of computational time i.e., computational time involved both in signcryption and unsigncryption, and the communication overhead or adding redundant bits to prove authenticity of the message is much smaller than that required by signature-then-encryption scheme as proved by Zheng.

A comparison of performance and cost involved using signcryption scheme is compared to well known sign-then-encrypt scheme like RSA, DSS combined with Elgamal encryption. Signcryption scheme can be implemented with a new algorithm and it may be possible to develop a better solution in terms of computation cost and communication overhead.

In the signcryption scheme of , the unsigncryption (decryption and signature verification) needs the recipient's private key (say x_b); therefore, only the recipient can verify the signature . The constraint of using the recipient's private key in unsigncryption is acceptable for certain applications where the recipient need not pass the signature to others for verification; however, Zheng's singcryption scheme cannot be used in applications where a signature need to be validated by a third party, through only the public key as in used in signature scheme. Here, to overcome this problem, we modify the Zheng's signcryption scheme so that verification of signature no longer needs sender's (x_a) and recipient's private key (x_b) by independent third party in case of any dispute. Hence, the modified scheme functions are exactly the same as that of signature–then– encryption approach. Modified scheme is approximately as efficient as Zheng's scheme and more computational and communicational efficient than the signature–then–encryption scheme. Now, we shall study some different signature-then-encryption scheme as well as some signcryption schemes [7-9].

II. DIFFERENT SIGNATURE-THEN-ENCRYPTION SCHEMES

A. Signature then encryption based on RSA :

Task: Alice has a message **m** to send to Bob. Alice has to use signature then- encryption **s**cheme by using RSA [4].

The RSA scheme is based on the difficulty of factoring large composite number. To use RSA, Alice and Bob has to choose public and private parameters as follows

p_a , q_a : large random prime numbers choose by Alice

$n_a = p_a \text{ X } q_a$

$\text{Phi}(n_a) = (p_{a-1}) \text{ X } (q_{a-1})$

Now Alice pick his public key y_a , so that gcd (y_a, Phi(n_a)) = 1; where $1 < y_a < \text{Phi}(n_a)$;

And calculate x_a, so that x_a X y_a mod Phi(n_a) = 1, i.e., x_a is the multiplicative inverse of y_a in mod Phi(n_a);

Alice public key (y_a, n_a) and private key x_a similarly Bob's public key (y_b,n_b) and private key is x_b;

- Signature generation: Alice generate signature s of message m in following steps :
  - i. Message m is HASH by any hash algorithm (eg SHA-1 or MD5), and generate Message digest MD1; i.e. MD1 = hash (m)
  - ii. Message digest MD1 is encrypting by Alice private key x_a by RSA algorithm and signature s is produced; i.e. s = $(MD1)^{x_a} \bmod n_a$
- Encryption : Encrypt message m , encrypt symmetric key k and then send to Bob:
  - i. c1 = E_k(m) ; Alice generate cipher text c1 of message m by using symmetric key k.
  - ii. c2 = $k^{y_b} \bmod n_b$; Alice encrypt symmetric key k (one time session key) by Bob's public key.
- Decryption : Decrypt c2 and then decrypt c1
  - i. k = $c2^{x_b} \bmod n_b$; now Bob have symmetric key k.
  - ii. m = D_k(c1); Bob decrypt cipher text c1 and produce message m.
- Signature verification: Bob can verify signature and authenticate it as he has received the message from Alice only. Alice also cannot refuse that he has not send it i.e. non repudiation is there.
  - i. Bob made HASH by any hash algorithm (which has been used by Alice at sender end) on message m which he has produce from decrement from c1 and made a message digest say MD2 ; i.e. MD2 = hash (m)
  - ii. Now s is decrypt by public key of Alice and get MD1; i.e. MD1 = $(s)^{y_a} \bmod n_a$
  - iii. Compare MD1 with MD2

If MD1 = MD2 => valid

If MD1 ≠ MD2 => not valid

B. Signature then encryption based on ElGamal :

Task: Alice has a message **m** to send to Bob. Alice has to use signature-then- encryption **s**cheme by using ElGamal [5].

The ElGamal scheme is based on the difficulty on hardness of computing descrete logarithm over a large finite field.To use ElGamal , Alice and Bob has to choose public and private parameters as follows

p : a large prime number

q : an integer in [1,..… ,p-1] with order p-1 modulo p.

g : an integer in [1,….,p-1] with order q modulo p. In practice, g is obtained by calculating $g = h^{(p-1)/q} \bmod p$. Here h is chosen uniformly at random from [2,….,p-1] and satisfies $h^{(p-1)/q} \bmod p > 1$.

x : is a random number from [1,…, p-1]. Here x must be chosen independently at random every time a message is to be signed by Alice. Here x is keep secret by Alice and x is chosen in such a way that x does not divide (p-1);

User Alice's private key is an integer $\mathbf{x_a}$ chosen randomly from [1,…..,p-1] with x_a does not divide(p-1), and her public key is $y_a = (g)^{x_a} \bmod p$;

User Bob's private key is an integer $\mathbf{x_b}$ chosen randomly from [1,…..,p-1] with x_b does not divide p-1), and her public key is $y_b = (g)^{x_b} \bmod p$;

- Signature generation: Alice generate signatures by generating of two numbers **r** and **s** on message m in following steps :

$r = (g)^{x} \bmod p$;

$s = (hash(m) - x_a . r) / x \bmod (p-1)$

- Encryption : By using Bob's public key , Alice can send him messages in a secure way.To do this, Alice chooses, for each message m, a random integer x , calculate symmetric key k:
  i. $k = y_b^{x} \bmod p$; calculation of symmetric key k ;
  ii. $c1 = E_k(m)$; Alice generate cipher text c1 of message m by using symmetric key k.
  iii. $c2 = g^{x} \bmod p$; This c2 is used to reproduce k at Bob's end by Diffie-Hellman key exchange methhod.
- Decryption : Decrypt c2 and then decrypt c1
  i. $k = c2^{x_b} \bmod p$; now Bob have symmetric key k.
  ii. $m = D_k(c1)$; Bob decrypt cipher text c1 and reproduce message m.
- Signature verification: Bob can verify signature and authenticate it as he has received the message from Alice only. Alice also cannot refuse that he has not send it i.e. non repudiation is there.
  i. Bob calculate h1 and h2 ah follows

$h1 = (g)^{hash(m)}$ and $h2 = (y_a^{r} . r^{s}) \bmod p$;

if h1 = h2 then (r , s) is regarded as Alice's signature on m .

C. Signature then Encryption based on "Schnorr Signature and ElGamal Encryption "

Task: Alice has a message **m** to send to Bob. Alice has to use signature-then- encryption **s**cheme by using Schnorr signature and ElGamal encryption [5-6].

Schnore signature scheme involves the following parameters:

Parameters Public key to all:

p : a large prime number

q : a prime factor of p-1.

g : an integer in [1,….,p-1] with order q modulo p. In practice, g is obtained by calculating $g = h^{(p-1)/q} \bmod p$. Here h is chosen uniformly at random from [2,….,p-1] and satisfies $h^{(p-1)/q} \bmod p > 1$.

Parameters specific to user Alice:

x_a : Alice private key chosen randomly from [1,….,q-1]

y_a : Alice Public key ; $y_a = (g)^{-x_a} \bmod p$.

x : is a random number from [1,…, q-1].

- Signature generation: Alice generate signatures by generating of two numbers **r** and **s** on message m in following steps :

$r = hash(g^{x} \bmod p , m)$; and $s = x + x_a . r \bmod q$

- Encryption : By using Bob's public key , Alice can send him messages in a secure way.To do this, Alice chooses, for each message m, a random integer x , calculate symmetric key k:

i. $k = y_b^x \bmod p$; calculation of symmetric key k ;
ii. $c1 = E_k(m)$; Alice generate cipher text c1 of message m by using symmetric key k.
iii. $c2 = g^x \bmod p$; This c2 is used to reproduce k at Bob's end by Diffie-Hellman key exchange method.

- Decryption : Decrypt c2 and then decrypt c1
  i. $k = c2^{x_b} \bmod p$; now Bob have symmetric key k.
  ii. $m = D_k(c1)$; Bob decrypt cipher text c1 and reproduce message m.
- Signature verification: Bob can verify signature and authenticate it as he has received the message from Alice only. Alice also, cannot refuse that he has not send it i.e. non repudiation is there.

Bob calculate and find out

$r = ((g^s . y_a^r \bmod p), m)$

if above are identical , then (r , s) is regarded as Alice's signature on m .

D. *Signature then Encryption based on "Digital Signature Standard (DSS) and ElGamal Encryption"*

Task: Alice has a message **m** to send to Bob. Alice has to use signature- then- encryption **s**cheme by using DSS signature and ElGamal encryption [5].

DSS signature scheme involves the following parameters:

Parameters Public key to all:

p : a large prime number

q : a prime factor of p-1.

g : an integer in [1,….,p-1] with order q modulo p. In practice, g is obtained by calculating $g = h^{(p-1)/q} \bmod p$. Here h is chosen uniformly at random from [2,….,p-1] and satisfies $h^{(p-1)/q} \bmod p > 1$.

Parameters specific to user Alice:

x_a : Alice private key chosen randomly from [1,….,q-1]

y_a : Alice Public key ; $y_a = (g)^{x_a} \bmod p$.

x : is a random number from [1,…, q-1].

- Signature generation: Alice generate signatures by generating of two numbers **r** and **s** on message m in following steps :

  $r = (g^x \bmod p) \bmod q$; and $s = (\text{hash}(m) + x_a . r) / x \bmod q$
- Encryption : By using Bob's public key , Alice can send him messages in a secure way.To do this, Alice chooses, for each message m, a random integer x , calculate symmetric key k:
  i. $k = y_b^x \bmod p$; calculation of symmetric key k ;
  ii. $c1 = E_k(m)$; Alice generate cipher text c1 of message m by using symmetric key k.
  iii. $c2 = g^x \bmod p$; This c2 is used to reproduce k at Bob's end by Diffie-Hellman key exchange methhod.
- Decryption : Decrypt c2 and then decrypt c1
  i. $k = c2^{x_b} \bmod p$; now Bob have symmetric key k.
  ii. $m = D_k(c1)$; Bob decrypt cipher text c1 and reproduce message m.
- Signature verification: Bob can verify signature and authenticate it as he has received the message from Alice only. Alice also cannot refuse that he has not send it i.e. non repudiation is there.

  Bob calculate and find out

  $r = ((g^{\text{hash}(m)} / ^{s.} . y_a^{r/s} \bmod p) \bmod q$

  if above are identical , then (r , s) is regarded as Alice's signature on m .

In all above cases of signature –then- encryption , if there is any dispute / non-repudiation, it may be resolved by third party, without compromising private keys of Alice (x_a) and Bob(x_b). Here , Alice send his public key y_a , and (m , s) in case of Signature-then-encryption based on RSA and (m ,r, s) in rest of the cases of Signature-then- encryption by Bob to third party. Following will be the steps to resolve the problem of non repudiation by independent third party, without compromising with sender and recipient's private keys [10 – 11].

$MD2 = \text{hash}(m)$; $MD1 = (s)^{y_a} \bmod n_a$;

If MD1 = MD2 valid and if $MD1 \neq MD2$ not valid

III. ZHENG SIGNCRYPTION SCHEME

Task: Alice has a message m to send to Bob [7-9]**.**

Public parameters

- p : a large prime.
- q: a large prime factor of p-1.
- g: $0 < g < p$ and with order q mod p.
- hash : 1 – way hash
- KH : key-ed one way hash .
- (E,D) : Private – key encryption and decryption algorithm .

Private parameters known to Alice:

Private key: x_a ; choose uniformly at random from [1, …. ,q-1];

Public key: $y_a = (g)^{x_a} \bmod p$;

Private parameters known to Bob:

Private key: x_b ; choose uniformly at random from [1, …. ,q-1];

Public key: $y_b = (g)^{x_b} \bmod p$;

Signcryption of message m by Alice the sender:

x : a number chosen uniformly random from (1,……, q-1).

Let $k = hash (y_b^{x} \bmod p)$; length of k is as per hash function chosen (128 bits or 160 bits);

Split k in two equal length k1 and k2. Use k1 for cipher text generation and k2 for signature generation.

$r = KH_{k2} (m)$;

$s = x / (x + x_a) \bmod q$;

$c = E_{k1}(m)$;

send to Bob the signcrypted text (c, r, s);

Unsigncryption of (c , r , s) by Bob the recipient:

The unsigncryption algorithm works by taking advantage of the property that $g^{x} \bmod p$ can be recovered by Bob from r , s, g , p . On receipt of (c , s , r) from Alice, Bob unsigncrypts as follows and reproduce k, by r , s, g , p, y_a and x_b;

$k = hash ((y_a . g^r)^{s \cdot x_b} \bmod p$;

Split k in two equal length k1 and k2 similar discipline as done by sender. Use k1 and k2 similar purpose as done by sender.

Decrypt cipher text c and reproduce plain text $m = D_{k1}(c)$;

Regenerate $r1 = KH_{k2} (m)$; if r = r1 then valid ; if $r \neq r1$ invalid signature.

Here, it is clear that to reproduce k at Bob (receiver) end x_b is directly involved. If there is any dispute / non-repudiation, then it cannot resolve by third party, without compromising private keys of Alice (x_a) and Bob(x_b).

IV. MODIFIED SIGNCRYPTION SCHEME

Proposed modified signcryption scheme perform task to transmit a message m with properties of correctness, efficiency, security (confidentiality, authentication, no repudiation). Its efficiency is better than available signature-then-encryption schemes and approximately same as Zheng's signcryption scheme with a third party authentication concept. Hence it is able to resolve any non reputation dispute.

Task: Alice has a message m to send to Bob.

Public parameters

- p : a large prime.
- q: a large prime factor of p-1.
- g: $0 < g < p$ and with order q mod p.
- hash : 1 – way hash
- KH : key-ed one way hash .
- (E,D) : Private – key encryption and decryption algorithm .

Private parameters known to Alice:

Private key: x_a; choose a prime number at random from [1, …. ,q-1];

Public key: $y_a = (g)^{x_a} \bmod p$;

Private parameters known to Bob:

Private key: x_b ; choose a prime number at random from [1, …. ,q-1];

Public key: $y_b = (g)^{x_b} \bmod p$;

Signcryption of message m by Alice the sender:

x : a number chosen a prime number at random from [1,……, q-1].

Generate Phi = x . x_a ; use for third party authentication or to resolve non repudiation problem if any dispute is there.

Let k = hash ($y_b^{x \cdot xa}$) mod p ; length of k is as per hash function chosen (128 bits or 160 bits);

Split k in two equal length k1 and k2. Use k1 for cipher text generation and k2 for signature generation.

r = KH_{k2} (m);

s = (x. x_a – r +1) mod q;

c = E_{k1}(m) ;

Send to Bob the signcrypted text (c, r, s);

Unsigncryption of (c , r , s) by Bob the recipient:

The unsigncryption algorithm works by taking advantage of the property that $g^{x.x}{}_a$ mod p can be recovered by Bob from r , s, g , p . On receipt of (c , s , r) from Alice, Bob unsigncrypts as follows and reproduce k, by r , s, g , p, y_a and x_b;

k = hash [y_a^{-1} . g^{s-r}] $^{x}{}_b$ mod p

Split k in two equal length k1 and k2 similar discipline as done by sender. Use k1 and k2 similar purpose as done by sender.

Decrypt cipher text c and reproduce plain text m = D_{k1}(c) ;

Regenerate r1 = KH_{k2} (m); if r = r1 then valid otherwise if r ≠ r1 invalid.

Here, it is cleared that to reproduce k at Bob (receiver) end x_b is directly involved. If there is any dispute / non-repudiation, then it can resolve by third party, without compromising private keys of Alice (x_a) and Bob(x_b).

Third party may get **(m , r , y_b)** from Bob and **Phi** from Alice to resolve dispute / non-repudiation . Here transmission of Phi, and (m , r , y_b) may be possible through unsecure public channel. Third party may generate k as follows

k = hash (y_b^{Phi} mod p);

split k in k1 and k2’ . generate r’ = hash (k2’ , m) and compare r with r’ for authentication.

Here , both sender and receiver do not compromise their Private key x_a and x_b for third party authentication , if ever required [7-9].

V. DISCUSSION

Following table showing computation cost and communication overhead of different signature-then-encryption schemes as well as Zheng and modified signcryption scheme.

TABLE : COMPUTATIONAL AND COMMUNICATION COST IN DIFFERENT SCHEMES

Various Schemes	Operations	Computational Cost	Communicational Overhead (in bits)
Signature-then-encryption based on RSA	Signature + Encryption	EXP=2,HASH=1,ENC=1	$\|n_a\| + \|n_b\|$
	Decryption + Verifying	EXP=2,HASH=1,DEC=1	
Signature-then-encryption based on ElGamal	Signature + Encryption	EXP=3,HASH=1,MUL=1,DIV=1,SUB=1,ENC=1	$\|q\|+ 2\|p\|$
	Decryption + Verifying	EXP=4,HASH=1,DEC=1	
Schnorr signature-then-encryption based on ElGamal	Signature + Encryption	EXP=3,HASH=1,MUL=1,ADD=1,ENC=1	$\|KH(.)\| + \|q\|+\|p\|$
	Decryption + Verifying	EXP=3,HASH=1,MUL=1,DEC=1	
DSS signature-then-encryption based on ElGamal	Signature + Encryption	EXP=3,HASH=1,MUL=1,DIV=1,ADD=1,ENC=1	$2\|q\| + \|p\|$
	Decryption + Verifying	EXP=3,HASH=1,MUL=1,DIV=2,DEC=1	
Signcryption scheme based on Zhengh	Signature + Encryption	EXP=1,HASH=1,DIV=1,ADD=1,ENC=1	$\|KH(.)\| + \|q\|$
	Decryption + Verifying	EXP=2,HASH=1,MUL=2,DEC=1	
Modified Signcryption scheme	Signature + Encryption	EXP=1,HASH=2,MUL=3,DIV=1,ENC=1	$\|KH(.)\| + \|q\|$
	Decryption + Verifying	EXP=2,HASH=1,MUL=1,DIV=2,DEC=1	

In all cases of signature-then-encryption, if there is any dispute / non-repudiation, it may be resolved by third party, without compromising private keys of Alice (x_a) and Bob (x_b). Alice public key y_a , is send by Alice and (m , s) in case of Signature-then encryption based on RSA and (m ,r, s) in case of Signature-then-encryption based on ElGamal , Signatur-then-Encryption based on "Schnorr Signature and ElGamal Encryption", Signature then Encryption based on "DSS Signature and ElGamal Encryption" will be send by Bob to third party. Following will be the steps to resolve the problem of non repudiation by independent third party, without compromising of sender and recipient's private keys.

MD2 = hash (m); MD1 = $(s)^{y_a} \bmod n_a$;

If MD1 = MD2 valid and if MD1 $\neq$ MD2 not valid

Zheng presented a positive answer to the following question : " is it possible to transfer a message of arbitrary length in a secure and authenticated way with an expense less than that required by signature-then-encryption? The proposed cryptographic primitive is more efficient for both cost: computational cost and communication overhead. It is determined by counting the number of dominant operations involved. The communication overhead represents the extra bits which are appended to a message in case of digital signature or encryption based on public key cryptography. With Zheng scheme, it is noted that to reproduce k at Bob (receiver) end x_b is directly involved and if there is any dispute / non-repudiation, then it cannot be resolved by third party, without compromising private keys of Alice (x_a) and Bob(x_b). Both the unforgeability and non-repudiation are based on the assumption that it is computationally infeasible to forge (m , r , s) (without knowing x_a , x_b) [4],[5],[7],[9].

The security of the modified scheme is the same as that of original scheme but it is computationally feasible for a third party to settle a dispute between Alice and Bob in an event where Alice denies that she is the originator of a signcrypted text. Third party may get **(m , r , y_b)** from Bob and **Phi** from Alice to resolve dispute / non-repudiation . Here transmission of **Phi**, and **(m , r , y_b)** may be possible through unsecure public channel. Third party may generate k by as follows

$k = hash (y_b^{Phi} \bmod p);$

Split k in k1 and k2' . generate r' = hash (k2' , m) and compare r with r' for authentication. Here , both sender and receiver do not compromise their Private key x_a and x_b for third party authentication , if ever required.

VI. CONCLUSION

In this paper we have evaluated the performance of available signature-then-encryption schemes, signcryption schemes. Proposed cryptographic primitive is more efficient for both types of costs involved: computational cost and communication overhead with third party authentication. It is also observed that proposed signcryption scheme has all the features like correctness, efficiency, security (confidentiality, authentication, non repudiation) and it is determined by counting the number of dominant operations involved. The communication overhead represents the extra bits which are appended to a message in case of digital signature or encryption based on public key cryptography. Proposed scheme is computationally feasible, for a third party to settle a dispute between Alice and Bob in an event where Alice denies that she is the originator of a signcrypted text. The reduction in computation and communication cost will result in fast and secure electronic communication.

REFERENCES

[1] William Stallings. *Cryptography and Network security: Principles and practices.* Prentice Hall Inc., second edition, 1999.

[2] Atul Kahate, *Cryptography and Network Security*, Tata McGraw-Hill Publishing Company Limited, year 2003.

[3] Behrouz A. Forouzan, *Cryptography & Network Security*, Tata McGraw-Hill Publishing Company Limited, year 2007.

[4] Rivest, R. Sharmir, A.,Adleman,L.A Method for obtaining digital signatures and public –key cryptosystem.Communications of the ACM,vol. 21 No. 2 1978,pp. 120-126.

[5] T.EIGamal,"A public key cryptosystem & a signature scheme based on discrete logarithms", IEEE transactions on information theory,IT-31(4):469-472,1985.

[6] C.P. schnorr, "efficient identification & signature for smart cards", Advances in cryptology-CRYPTO'89,LNCS 435, SPRINGER-VERLAG,pp. 239-251.

[7] Yuliang Zheng "Signcryption and Its Applications In Efficient Public Key Solutions " Lecture Notes in computer Science, Vol. 1397, pp. 291-312,Springer- Verlag,1998.

[8] H. Petersen & M.MICHELS, "cryptanalysis & improvement of signcryption schemes", to appear in IEE computers & digital techniques, 1998.

[9] Yuliang Zhen, Digital signcryption or how to achieve cost (signature and encryption) << cost (signature) + cost (encryption). In CRYPTO '97: Proceeding of the 17th Annual International Conference on Advances in Cryptology, pages 165-179, London, UK, 1997. Springer-Verlag.

[10] Zheng Peng , Jia Jian Fang, "Comparing and implementation of Public Key Cryptography Algorithms on Smart Card", IEEE ,2010 International conference on Computer Application and System Modeling (ICCASM 2010) , vol. 12,pp. 508-510..

[11] Lanxiang Chen , Shuming Zhou, "The Comparisons between Public key and Symmetric key Cryptography in Protecting Storage Systems", IEEE, 2010 International conference on Computer Application and System Modeling (ICCASM 2010) , vol. 4,pp. 494-502.

AUTHORS PROFILE

Bharat Gupta received the Bachelor degree in Computer Science and Engineering from Amravati University,India in 1992and Post Graduate degree from Rajasthan Technical University, India in 2009.He is a research student of Mewar University, Chittorgarh Rajasthan , India.He has total 20 years of experience(8 years in industry and 12 years of teaching).His interests are in Information security and Database management system.

Dr. Rajeev Gupta, received M.Tech. and Ph.D. degree from Indian Institute of Technology Mumbai, India. He is currently working as Professor and Head of Electronics and Communication Engineering department ,University College of Engineering, Rajasthan Technical University, Kota, Rajasthan, India. He has 25 years of teaching and research experience. He has several publications in reputed International journals and conferences. His main research interests include soft computing, network security and intelligent control.