Middle-East Journal of Scientific Research 22 (12): 1746-1752, 2014
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2014.22.12.21866

# Keccak MD Hash Algorithm Based Tag Kem for Certificateless Hybrid Signcryption

[1,2]*R. Sujatha and* [1]*M. Ramakrishnan*

[1]Velammal Engineeering College, Chennai-66
[2]Madurai Kamrajar University, Madurai, India

**Abstract:** This paper presents a novel idea to implement Key Encapsulation Mechanism (KEM) in a CL hybrid encryption setting to provide security to the key itself, before the actual data encryption of long message. Tag-KEM (Key Encapsulation Mechanism with a tag) is an authenticated method for generic construction of hybrid encryption. Tag- KEM (TKEM) allows the sender to encapsulate a symmetric key along with a tag so that the receiver can authenticate the sender, the key and the tag. This paper deals with the proposal of an effective model for KEM in the first phase and applying the concept of Tag as well as introducing a new hashing mechanism known as Keccak-MD Hashing in the second phase. This is the first certificateless hybrid encryption which uses Keccak algorithm for hashing. The symmetric key designed is then used to encrypt long message in DEM (Data Encapsulation Mechanism). The construction outperforms the existing methods in various aspects as explained in the following sections. It is subsequently proved that this model is one-way-CCA secure, which provides insider and outsider security, also secure against cube attacks. Basic public key encryption schemes have limited message spaces whereas the proposed ciphers can cope with long messages and are not slower than traditional asymmetric ciphers.

**Key words:** Certificateless Hybrid Encryption · Chosen Ciphertext Security · Cube Attacks · Kecakk Hashing · Tag KEM.

## INTRODUCTION

Encryption and signature schemes are fundamental cryptographic tools for providing privacy and authenticity in the public-key setting. Both privacy and authenticity are simultaneously needed in many applications on ad-hoc network where anyone can freely join or leave the network. Signcryption is an asymmetric type of cryptographic technique which provides simultaneously both message confidentiality and unforgeability at a low computational cost and communication delay. Signcryption first proposed by Zheng [1], is a cryptographic primitive that fulfills both the functions of digital signature and public key encryption simultaneously, at a cost significantly lower than that required by the traditional signature-then-encryption approach. Through Signcryption one can achieve simultaneously confidentiality, authenticity and non-repudiation of transmitted data.

But there is a need to provide an assurance to the user about the relationship between a public key and the identity of the user of the corresponding private key. Al-Riyami and Paterson[2] introduced a new paradigm called certificateless cryptography. The certificateless cryptography does not require the use of certificates and yet does not have the built-in key escrow feature of IBC.The certificateless setting eliminates the certificate authority as in traditional PKI and Key escrow problem in IBC. The proposed model imposes the receiver to compute a private key for itself and transmit the public key to the sender. The essential security requirement for digital signatures is the unforgeability against adaptive chosen message attacks [3], where an attacker is allowed to query a number of messages of his choice. Security in CCA attacks means that an adversary obtains no information about encrypted messages. Security against chosen-ciphertext attack is equivalent to the notion of non-malleability.

**Prelimaniries:** To simplify key management procedures of traditional PKI, Shamir proposed the concept of identity-based cryptography (IBC) [4]. The idea of IBC is

**Corrsponding Author:** R. Sujatha, Velammal Engineeering College, Chennai-66, India.

to get rid of certificates. The user's public key to be any binary string that uniquely identifies the user. IBC uses a trusted third party called private key generator (PKG). The PKG generates the secret keys of all of its users, so a user can decrypt only if the PKG has given a secret key, the dependence on the PKG who can generate all users' private keys inevitably causes the key escrow problem to the IBC.

**Hybrid Signcryption:** Public key encryption schemes often limit the message space to a particular group, which can be restrictive when one wants to encrypt arbitrary messages. For this purpose, hybrid schemes are devised. Hybrid certificateless encryption scheme encrypts message of unbounded length [5].Tag-KEM uses asymmetric technique to encrypt a symmetric key along with a tag, while the DEM uses a symmetric cipher to encrypt the message payload using the key from the KEM. In this method a symmetric encryption scheme is used to overcome the problems associated with encrypting long messages using "pure" asymmetric techniques. This is achieved by encrypting the message with a symmetric encryption scheme and a randomly generated symmetric key using PRNG generator. This random symmetric key is then encrypted using an asymmetric encryption scheme. Hybrid encryption scheme can be split into two distinct components: an asymmetric key Encapsulation Mechanism (KEM) and a symmetric Data Encapsulation Mechanism (DEM). Data encapsulation mechanism (DEM) takes a key k and a message m and computes c = DEM k(m). Given k, one can recover m =$DEM^{-1}$k (c). The key k is transferred to the in the form of encapsulation. To create encapsulation, the sender uses a key encapsulation mechanism (KEM). This is an algorithm which takes as input a public key $P_K$ and outputs a session key k plus an encapsulation c' of this session key. (k,c')?KEM(PK).The recipient recovers the key k using his private key $S_K$ using the decapsulation mechanism. k?KEM-1(c', $S_K$). The full ciphertext of the message m is then given by (c',c).

In this paper we have discussed about implementing KEM using Keccak algorithm which is CCA-secure and provides security against cube attacks There are several examples of hybrid encryption schemes that do not fit into the KEM/DEM model.The Encryption scheme in DEM can use the secret information (a key) to encode a message in a way that an eavesdropper cannot decode it.

Hybrid Signcryption introduced by Shoup [6] combines a key encapsulation mechanism (KEM) and a data encryption mechanism (DEM). While it is sufficient

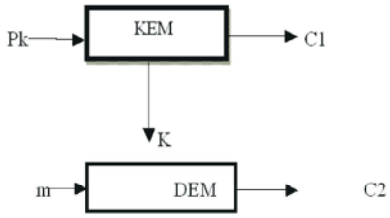

Fig. 1: Encryption algorithm

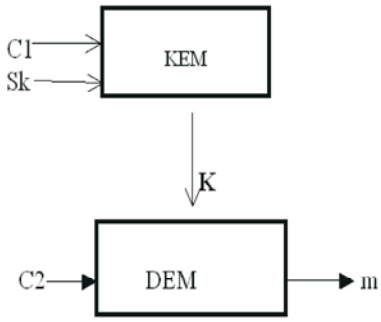

Fig. 2: Decryption algorithm

to require both components to be secure against chosen ciphertext attacks [7] (CCA-secure), Kurosawa and Desmedt [8] presented a particular example of KEM that is not CCA-secure but can be securely combined with a specific type of CCA-secure DEM to obtain a more efficient, CCA-secure hybrid encryption scheme. The scheme proposed is more efficient than existing techniques.

**CL-KEM:** A KEM consists of the following three algorithms: (I) a key-pair generation algorithm KEM. K Gen (ii) a key encryption algorithm KEM. Enc, (iii) a key decapsulation algorithm KEM. Dec.

- *Master Key Gen:* On input k where k = 1 to n is a security parameter, it generates a master public/private key pair (mpk, msk) [9, 10].
- *Partial KeyGen:* On input msk and a user identity ID it generates a user partial key pskID.
- *User KeyGen:* On input mpk and a user identity ID, it generates a user public/private key pair (upkID, uskID).
- *Enc:* On input mpk, a user identity ID, a user public key upkID and a message m, it returns a ciphertext C1.
- *Dec:* On input a user partial key pskID, a user private key uskID and a ciphertextC1, it returns the plaintext m or indicates the failure of decryption -.

**CL-TKEM:** Tag-KEM is a KEM with a tag. The Encapsulation algorithm of a KEM is split into two sections in a Tag-KEM method, Key Generation and Encapsulation. Key generation remains the same in a Tag-KEM as CL-KEM and Decapsulation is modified to take a tag as an additional input. TKEM generation is a probabilistic algorithm that generates public key pk and private key sk. pk is used to encapsulate a session key and the decapsulation is done through sk. pk is a probabilistic algorithm that generates a session key K and internal state information w. The session key is used for encryption in DEM. Enc (w, $\tau$) is a probabilistic algorithm that encrypts K to C, using $\tau$, where $\tau$ is a tag. The significance of choosing TKEM is that KEM has to encapsulate random strings and may generate them by itself, where as ordinary encryption scheme has to encrypt any strings given as input [11].

CL-TKEM consists of the following six steps:

- *CL-KEM Setup:* On input $1^k$ where k-N is a security parameter, it generates a master public/private key pair (mpk; msk).
- *CL-KEM partial KeyDerivation:* On input msk and a user identity ID it generates a user partial key / ID-based private key skID.
- *CL-KEM User KeyGen:* On input mpk and a user identity ID, it generates a user public/private key pair (upk; usk).
- *CL-KEM Key Verification:* On input mpk and upk, it generates an encryption key enck that is used for all following encapsulations. This algorithm needs to run only if the master public key or the user public key change (which should happen less frequent than actual encapsulations take place).
- *CL-KEM Encapsulation:* takes as input (mpk; enck; ID) and outputs an encapsulation key pair (K;C) where C is called the encapsulation of the key K respectively. Encap(w, $\tau$).
- *CL-KEM Decapsulation:* takes as input (skID; usk); ID; C) and decapsulate C to get back a key K, or outputs the special Symbol 'e' indicating invalid encapsulation. Decap (pskID, uskID, ID, $\tau$, e).

Now Key encapsulation mechanisms (KEM) provide efficient means to communicate a random key from a sender to a designated receiver. This key is later used in DEM.

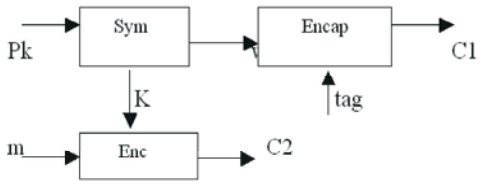

Fig. 3: Existing System

In the Existing system[12] Symmetric key encryption transmitted on wire is prone to get held by the intruder. Hash mechanism of tag is not collision free. Insider security cannot be achieved. Key encapsulation part of KD scheme by itself is not CCA2 secure.

**Proposed Keccak Hash Function:** In this paper we have shown how to construct a hybrid certificateless encryption scheme to encrypt messages of unbounded length, by using the CL-TKEM using Keccak Algorithm. The Advanced Hash Standard Keccak- SHA 3 algorithm announced by NIST is an elegant and convincing design which is less complex than existing message digest algorithms and stronger as the SHA algorithm. Keccak hash function supports the same hash lengths as SHA-2 and its internal structure differs significantly from the rest of the SHA family. Hash algorithms are used widely for cryptographic applications that ensure the authenticity of digital signatures. These algorithms take an electronic file and generate a short digest. Any change in the original message, however small, must cause a change in the digest and for any given file and digest, it must be infeasible for a forger to create a different file with the same digest. Keccak is safe against cube attacks. These type of attacks are leakage attacks in which only a single bit of information in each encryption is available to the cryptanalyst. Almost all the other cryptanalytic techniques require knowledge of big chunks of data in order to partially encrypt or decrypt them. The attacker can find (via physical probing, power measurement, or any other type of side channel) one bit of information about the intermediate state of the encryption after each round. An interesting property of cube attacks is that they can be applied even when this polynomial is completely unknown (e.g., when the attacker probes a random wires in a dense chip and does not know which signal it carries). Since the bits computed during the early rounds can be typically represented by low degree multivariate polynomials, cube attacks seem to be an ideal generic key recovery technique in these

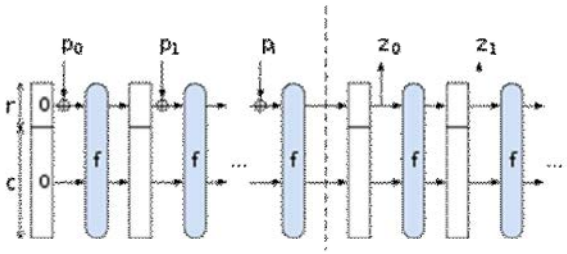

Fig. 4: Sponge construction of Keccak Hash fubnction

situations. Keccak algorithm is known for its elegant design and its ability to run well on many different computing devices. The clarity of it's construction lends itself to easy analysis and has higher performance in hardware implementations than SHA-2. Keccak has the added advantage of not being vulnerable. The algorithm uses the sponge construction in which message blocks are XORed into the initial bits of the state. The sponge function is a cryptographic hash function with infinite output and can perform all symmetric cryptographic functions, from hashing to pseudo-random number generation.

In the above diagram $p_i$ are input, $z_i$ are hashed output. The unused "capacity" c should be twice the desired resistance to collision . To absorb *r* bits of data, the data is XORed into the leading bits of the state and the block permutation is applied. First r bits of the state are produced as output and the block permutation is applied if additional output is desired. The number of message bits processed per block permutation *r*, depends on the output hash size. To compute a hash, initialize the state to 0, pad the input and break it into *r*-bits. Absorb the input into the state, XOR it into the state and then apply the block permutation. Benefits of Keccak are, it has arbitrary output length this allows to simplify modes of use where dedicated constructions would be needed for fixed-output-length hash functions and Keccak excels in hardware performance, with speed/area trade-offs and outperforms SHA-2.

**Security Proof:** This paper focuses on implementing TKEM module using Keccak MD hash algorithm. This model provides full insider security means that if the sender or receiver private key is exposed, an attacker will still be not able to recover the message from the ciphertext. A parameter unknown to the insider and outsider is used for computing the actual symmetric key. As only a symmetric key parameter is encapsulated with a tag and transferred to the receiver, the time complexity for the intruder is increased. The symmetric key for signcryption is obtained from the hash output. To address the problem of cube attacks new TKEM technique has been developed using Keccak SHA-3 hash algorithm.

The design of the tag- KEM is done such a way that it addresses three attacks and provides a solution in order to make the probability of a passive attack to a negligible value. The first parameter to be considered is the chosen ciphertext attack (CCA), which is eliminated by the use of session key so that no pattern drawn by the intruder helps in obtaining the message. The notion of chosen-cipher text security means, even if the adversary is allowed to query a decryption oracle on cipher texts of his choosing, he obtains no information about messages encrypted in other cipher texts.

The second is the outsider attack. The Keccak-MD hashing algorithm proposed here is dependent on a factor decided by the network administrator, which is a periodic value. Hence an outsider finds it time consuming to crack the message and thus the session gets completed. In addition to this, the hashing algorithm is proposed with effective round functions which prove to be better than existing message digest hash algorithms.

As this tag-KEM model implements hashing only at the sender and receiver side, there is no point of collision coming into existence. The main design requirements for the hashing techniques are it is difficult for an enemy to find two inputs that hash to the same result (collision resistance). Given a hash, find an input that gives that result (pre-image resistance), given an input, find another input that hashes to the same result (second pre-image resistance). Proposed TKEM is IND-CCA2 secure if Adversary A (pdec, t) is negligible where pdec is no. of decapsulation queries and t is running time of A. Negligible Function is a function f: N?R is called negligible if for every positive polynomial.

P there exists a k0 such that for all k > k0 we have
f (k) <1/ P(k).

Finally, the insider attack is handled by the addition of an offset to the random number before hashing. This offset value is stored in an access restricted file in the shared memory and thus, only the sender and receiver have access to this file. This eliminates the possibility of attack by an untrusted node within a network.

**Proposed TKEM Model Using KECCAK MD Hashing (KMD):**

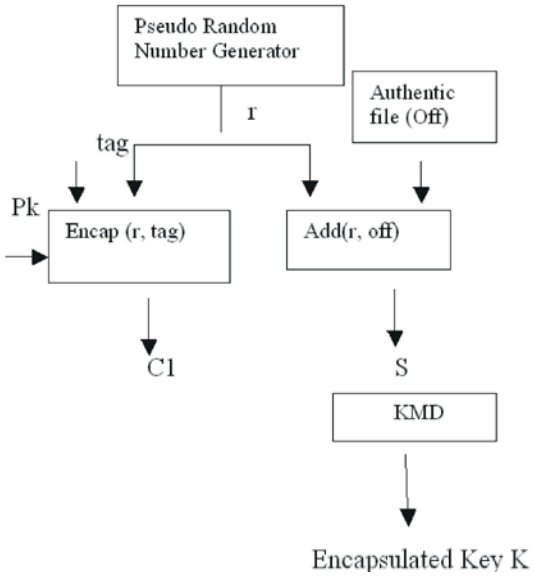


Fig. 5: Proposed CL-TKEM model- Sender Side


Fig. 6: Proposed CL-TKEM model- Receiver Side

Encapsulated Key K

**Asymmetric Key Generation:**

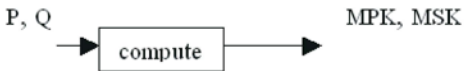


Fig. 7: Key Generation for the proposed system

compute n = p.q
calculate ø (n)
ø (n) = ø (pq)
= ø (p). ø (q)
= (p-1) (q-1)

Select e such that it is relatively prime to ø(n) as well as gcd (ø (n), e ) = 1

Find d such that d.e = 1 (mod ø (n)
Thus, Public Key mpk = { e, n }
Private Key msk = { d, n }

**KECCAK-MD Hash Algorithm:**

*Step 1:* Convert the input to binary digits. Now obtain the length L of the binary string M.

*Step 2:* Append '1' followed by '0' till the new length becomes congruent to 448 mod 512.

*Step 3:* Convert the value L into 64 bit binary and append to M. Now the length of message is N.

*Step 4:* Apply the round function of keccak MD hash to M in 16 bit block, as per the following pseudo Code.

*Step 5:* Initialize buffer

X = 10234567
Y = 89ABCDEF
Z = FEDCBA98
W = 76543201

*Step 6:* Process each 16 bit block

/ * Process each 16-word block. * /
For i = 0 to N/16-1 do
/ * Copy block i into P * /
For j = 0 to 15 do
Set P[j] to M [i*16+j].
End
/ * of loop on j * /
/ * Save X as XX, Y as YY, Z as ZZ and W as WW * /

XX = X
YY = Y
ZZ = Z
WW = W

/ * Round 1. * /
/ * Let [xyzw k s] denote the operation * /
x = (x + F(y,z,w) + M[k]) <<< s.
/ * Do the following 16 operations. * /

[XYZW 0 2] [WXYZ 1 6]
[ZWXY 2 10] [YZWX 3 18]
[XYZW 4 2] [WXYZ 5 6]
[ZWXY 6 10] [YZWX 7 18]
[XYZW 8 2] [WXYZ 9 6]
[ZWXY 10 10] [YZWX 11 18]
[XYZW 12 2] [WXYZ 13 6]
[ZWXY 14 10] [YZWX 15 18]

/ * Round 2. * /
/ * Let [xyzw k s] denote the operation * /

x = (x + G(y,z,w) + M[k] + 5A827999) <<< s / * Do the following 16 operations. * /

[XYZW 0 2] [WXYZ 4 4

[ZWXY 8 8] [YZWX 12 12]
\[XYZW 1 2] [WXYZ 5 4]
[ZWXY 9 8] [YZWX 13 12]
[XYZW 2 2] [WXYZ 6 4]
[ZWXY 10 8] [YZWX 14 12]
[XYZW 3 2] [WXYZ 7 4]
[ZWXY 11 8] [YZWX 15 12]

/ * Round 3. * /
/ * Let [abcd k s] denote the operation * /

x = (x + H(y,z,w) + M[k] + 6ED9EBA1) <<< s.
/ * Do the following 16 operations. * /
[XYZW 0 2] [WXYZ 8 8]
[ZWXY 4 10] [YZWX 12 14]
[XYZW 2 2] [WXYZ 10 8]
[ZWXY 6 10] [YZWX 14 14]
[XYZW 1 2] [WXYZ 9 8]
[ZWXY 5 10] [YZWX 13 14]
[XYZW 3 2] [WXYZ 11 8]
[ZWXY 7 10] [YZWX 15 14]

/ * perform the following additions. (That is, increment each of the four registers by the value it had before this block was started.) * /
X = X + XX
Y = Y + YY
Z = Z + ZZ
W = W + WW
end / * of loop on i * /

*Step 7:* Repeat step 4 for n times. / * n decided by the local network administrator */
*Step 8:* The symmetric key K is the 128 bit output of the hash algorithm obtained as XYZW.

**Intrusion Time Comparison:** The graph shows the time taken for the intruder to crack the message in all the three systems (CL- KEM, Existing Tag KEM and proposed KMD Tag KEM). It is evident from the graph that the proposed tag-KEM makes the intruder to consume indefinite time in order to crack the message[13-25].

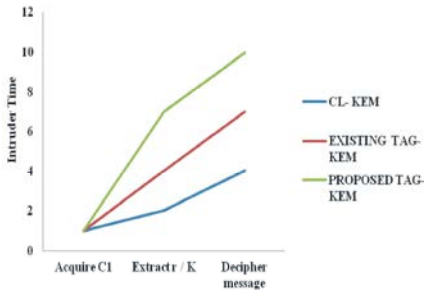

Fig. 8: Intrusion Time Graph

Table 1: Comparison with Existing System

| Algorithm | No. of bit operations in Round Function |
|---|---|
| MD 4 | 12 |
| MD 5 | 14 |
| Keccak MD | 11 |

## CONCLUSION

The paper has proposed a new method to implement TKEM in hybrid signcryption to provide high security and performance for ad-hoc network based applications where anyone can freely join or leave the network. The main advantage of the KEM-DEM technique over direct use of PKE is encryption of large messages is performed faster and a tighter security reduction. The proposed system foregrounded a pioneering technique in the certificateless signature scheme that covers the property of strong unforgeability in the standard model. In future research work, Tag KEM can be implemented using True Random number generator for tight security.

## REFERENCE

1. Zheng, Y., 1997. Digital Signcryption or How to Achieve Cost(Signature and Encryption) Cost (Signature) + Cost (Encryption), Crypto '97, LNCS. Available. www.ijcaonline.org., 1294: 165-179.
2. Al-Riyami, S.S. and K.G. Paterson, 2003. Certificateless public key cryptography. in *Advances in Cryptology-ASIACRYPT,* LNCS 2894, Springer-Verlag, pp: 452-474.
3. Goldwasser, S., S. Micali and R. Rivest, 1988. A Digital Signature Scheme Secure Against Adaptive Chosen Message Attacks, SIAM, Journal on Computing, 17(2): 281-308.
4. Shamir, A., 1984. Identity-based cryptosystems and signature schemes. In Proc. of CRYPTO 84, volume 196 of Lecture Notes in Computer Science, Springer-Verlag, pp: 47–53.
5. Cramer, R. and V. Shoup, 1998. A practical public key cryptosystem Provably secure against adaptive chosen cipher text attack, In CRYPTO '98, LNCS 1462,. Springer-Verlag, pp: 13-25.
6. Shoup, V. and R. Gennaro. Securing threshold cryptosystems against chosen cipher text attack. In EUROCRYPT '98, LNCS Springer- Verlag. Available. http://shoup.net/iso, 1403: 1-16.

7. Kurosawa, K. and Y. Desmedt, 2004. A new paradigm of hybrid encryption scheme. In M. Franklin, editor, Advances in Cryptology-crypto 2004, volume 3152 of Lecture Notes in Computer Sciene, Springer-Verlag, pp: 426-442.
8. Okamoto, T., S. Uchiyama, and E. Fujisaki, 1999. EPOC: Efficient Probabilistic public-key encryption. Submission to Standard Specifications for Public-Key Cryptography, Additional techniques, 13653.
9. Dodis, Y. and Signcryptio, 2005. Encyclopedia of Cryptography and Security, Springer Verlag.
10. Abe, M., R. Gennaro and K. Kurosawa, 2005. Tag-KEM/DEM: A new framework for hybrid encryption. IACR ePrint Archive, 2005/027.
11. Kurosawa, K. and Y. Desmedt, 2004. A new paradigm of hybrid encryption scheme. In M. Franklin, editor, Advances in Cryptology Crypto 2004, of Lecture Notes in Computer Science, Springer-Verlag, 3152: 426-442.
12. Bellare, M., A. Desai, D. Pointcheval and P. Rogaway, 1998. Relations among notions of security for public-key encryption schemes. In HK rawczyk, editor, CRYPTO'98, volume 1462 of LNCS, Springer-Verlag, 26: 45.
13. Bjørstad, T.E. and A.W. Dent, Building Better Signcryption Schemes with Tag-KEMs, URL: http://eprint.iacr.org/2005/405.pdf.
14. Boneh, D. and J. Katz, 2008. Improved efficiency for CCA-secureCrypto systems built using identity-based encryption. IACR ePrint archive, 2004/261.
15. Dent, A.W., 2005. Hybrid signcryption schemes with insider security. In Proceedings of ACISP 2005, volume 3574 of Lecture Notes in Computer Science,.Springer Verlag, pp: 253-266.
16. Fujisaki, E. and T. Okamoto, 1999. Secure integration of asymmetric and symmetric encryption schemes. In M. Wiener, editor, Advances in Cryptology Crypto '99, volume 1666 of Lecture Notes in Computer Science,. Springer-Verlag, pp: 535-554.
17. Barbosa, M. and P. Farshim, 2008. Certificateless Signcryption. ACM Symposium on Information, Computer and Communications Security ASIACCS.
18. Chen, L. and Z. Cheng. Security Proof of Sakai–Kasahara's Identity Based Encryption Scheme. Cryptography and Coding, LNC, 3796: 442–459, Springer-Verlag.
19. Chen, L., Z. Cheng, J. Malone-Lee and N.P. Smart, 2006. An Efficient ID KEM Based on the Sakai-Kasahara Key Construction, IEEE Proceedings Information Security, 153: 19-26.
20. Bleichenbacher, D., 1998. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In CRYPTO '98, LNCS, 1462: 1-12.
21. Goldwasser, S., S. Micali and R. Rivest, 1988. A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks, SIAM Journal on Computing, 17(2): 281-308.
22. Fujisaki, E. and T. Okamoto, 1999. Secure integration of asymmetric and symmetric encryption schemes. In M. Wiener, editor, Advances in Cryptology Crypto '99, volume 1666 of Lecture Notes in Computer Science, Springer-Verlag, pp: 535-554.
23. Okamoto, T., S. Uchiyama and E. Fujisaki. EPOC: Efficient Probabilistic public-key encryption. Submission to P1363a: Standard Specifications for Public-Key Cryptography, Additional Techniques, pp: 999.
24. Boneh D. and M. Franklin, 2003. Identity-Based Encryption from the Weil Pairing.SIAM Journal on Computing, 32:586–615
25. V. Shoup., 2000. Using hash functions as a hedge against chosen cipher text attack. In Proc. of EUROCRYPT 2000, volume 1807 of Lecture Notes in Computer Science, Springer-Verlag, pp: 275–288.