ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering and Technology, Madurai, Tamil Nadu, India

TRNG Based Key Generation for Certificateless Signcryption

Alagarganesh Thangathiruppathi, S.M.Udhaya Sankar

Student, Department of Information Technology, Velammal Engineering College, Chennai, India.

Assistant Professor, Department of Information Technology, Velammal Engineering College, Chennai, Tamilnadu, India.

Abstract - Signcryption is a cryptographic primitive that fulfills both the functions of digital signature and public key encryption simultaneously in low cost compare with the traditional signature-then-encryption approach. In this paper, we propose a new method for providing more secure key generation against the outer and insider attack, which is based on the mouse movements. Tag KEM process is implemented using this True random number generator (TRNG) and it generates most secure and non-deterministic secret keys for data encryption which held in the Data encapsulation mechanism (DEM). We compare the statistical reports of the proposed system with the previous methods which is used to implements TKEM based on pseudo random number generators, and it works better in manner of security.

Keywords - Signcryption, Tag Key Encapsulation Module, True Random Number Generators.

I. INTRODUCTION

In earlier stage, most of the communication using signature and encryption approach. In these approach first unique signatures is generated according to the sender of the message. Next sending message is encrypted by randomized key which is generated by some algorithm and it provides cipher text of the sending message. Finally encrypt the randomized key value using receiver's public key which is useful in decryption process. The main disadvantage of these approach is large computation time, power and cost. Signcryption is a new cryptography primitive which is discovered by Yuliang Zheng in 1997. Signcryption is a new paradigm in public key cryptography that simultaneously fulfills both the functions of digital signature and public key encryption in a logically single step, and with a cost significantly lower than that required by the traditional "signature and encryption" approach [1].

A signcryption scheme typically consists of three algorithms: Key Generation (Gen), Signcryption (SC), and Unsigncryption (USC).Key Gen generates a pair of keys for any user who is participated in signcryption, SC is generally a probabilistic algorithm which is used to generate the cipher text of corresponding plaintext with the help of public and private keys, and USC is most likely to be deterministic one which is used to decrypt the plain text from cipher text.

Signcryption is a combination of two schemes; one of digital signatures and the other of public key encryption. One can implement Signcryption by using any Digital signature schemes like ElGamal's shortened digital signature scheme, Schnorr's scheme in conjunction with a public key encryption scheme like DES, 3DES or SPEED [2]. This choice would be made based on the level of security desired by the users. Here we consider the implementation of Signcryption using ElGamal's shortened signature scheme and a public key encryption algorithm denoted by Enc and Dec (Encryption and Decryption algorithms) [2].

The parameters involved in signcryption are: -

m – The message
a – A large prime number
b – A large prime factor of p.
g – An integer with order q modulo p chosen randomly from the range 1... a-1
x – a number chosen uniformly at random from the range 1,…,b-1
Xa – Alice's private key chosen randomly from the range 1... a-1
Ya – Alice's public key ya = gXa mod a.

First step is to compute r, which is a hash of the message m with additional parameters involved.

Next we compute the component, s, using Alice's private key which used as secret key of the sender's side to generate encrypt key for the plaintext encryption. We do this as follows: -

Next these two components, (r and s) are sent to Bob, along with the message m. On receiving this, Bob uses r, s and Alice's public key to obtain the value k. Then he

does a hash of the message using k and verifies that it is equal to r. Bob accepts the message only if the hash of m and k matches. It ensure that Alice digitally sign the message. The other scheme involved in the Signcryption algorithm is public key encryption. The Following steps can explain about it: -

Cipher text = enc (plaintext, PK).

Plaintext = dec (cipher text, PK-1).

Where PK is the public key and PK-1 is the private key. In signcryption you have to generate public keys which are used in encryption based on the private key of that user which is selected from random parameter space.

II. RELATED WORK

In 1997 Zhang introduced signcryption primitive and proposed security model and notions for that model [1]. But it could not contain any features to authenticate the non-reputation of the users. In 2002 Malone lee proposed an identity based signcryption scheme which is based on identity based cryptography introduced by Shamir in 1984[11]. It provides first security model and notions for identity based signcryption. In identity-based cryptography [4] an arbitrary bit-string representing a user's identity can be used as the encryption or verification of public key. This means that public key certificates are not required. This feature, however, comes at the cost of introducing an all-powerful secret key issuing authority, which authenticates users and provides secret keys through a secure channel. The problem is that, not only must this authority be trusted to authenticate the users, but also not to take advantage of possessing the user's secret keys. This is known as the key escrow property of identity-based cryptosystems.

In 1985, certificate less cryptography was introduced by S.S. Al-Riyami and K.G. Paterson [5].In certificate less schemes, key escrow is seen as an undesirable property, and user encryption and verification keys contain both a user identity and a public key which is unauthenticated.

Correspondingly, secret key of users generated from two partial secret values: one coming from an identity-based trusted authority called the Key Generation Centre (KGC) and another one generated by the user.[6] Certificate less security models capture scenarios where the attacker can be a system users known as outsider or KGC itself known as insider. Consider that User public keys are not authenticated; attackers are allowed to replace user's public keys to attempt impersonation. Since numerous certificate less encryption and signature schemes, and variants thereof, have been proposed.

Wenjian Xie and Zhang Zhang, [2009] proposed concrete constructions of certificate less signcryption in the literature are built from bilinear maps and presented a pairing-free certificate less signcryption scheme, which is more efficient than all other constructions Organization[8].

Fagen Li, Masaaki Shirase, and Tsuyoshi Takagi, [2009] addressed about the possibilities to construct a hybrid signcryption scheme in the certificate less setting [3]. They extend the concept of signcryption tag-KEM to the certificate less setting and show how to construct a certificate less signcryption scheme using certificate less signcryption tag-KEM. They also give an example of certificate less signcryption tag-KEM.

S. Sharmila Deva Selvi, S. Sree Vivek and C. Pandu Rangan,[2008] proposed an improved certificate less hybrid signcryption scheme and formally prove the security of the improved scheme against both adaptive chosen cipher text attack and existential forgery in the appropriate security models for certificate less hybrid signcryption[7]**.**

P.Murali and R.Palraj, [2011] proposed a new method for generating True random numbers based on image which generates 256 bits key or higher for key exchange algorithm[9]. True random numbers are always secured and good, compared to pseudo random numbers. The attacker could not derive the key from the image and also it should not require any additional devices. But Small changes in the image should leads to a significant difference in the generated random number so communication noise should play important role in information retrieving.

III. 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, τ) is a probabilistic algorithm that encrypts K to C, using τ, where τ 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.

CL- TKEM consists of the following six steps.

1. CL-KEM **Setup**: On input 1^k where k € N is a security parameter, it generates a master public/private key pair (mpk; msk).
2. CL-KEM **partial Key Derivation**: On input msk and a user identity ID it generates a user partial key / ID-based private key skID.
3. CL-KEM **User KeyGen:** On input mpk and a user identity ID, it generates a user public/private key pair (upk; usk).
4. 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).
5. 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, τ).
6. 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, τ, 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.

IV. PROPOSED SYSTEM

Consider mouse as a proper source for TRNs, because it is used commonly and it produces a real random source. People move mouse in a unique random pattern, and others could not find out the patterns. Even the attackers get the pattern of the events, they cannot predict the future values of the user's mouse pattern. We propose a true random number generator using mouse movement for generating Master pairs used in Signcryption.

In this work, the *x*-coordinate of the mouse position to be the length of an iteration segment of TRNs and the *y*-coordinate to be the initial value of this iteration segment. It is a simple algorithm of TRNGs based on user's mouse movement. Therefore, we can conclude that our algorithm is an effective and practicable method to produce TRNs for universal computers.

Based on this set pairs generating the private key for the system with the help of KEM. Generation of secret key and encrypting it using a public key encryption scheme is called key encapsulation mechanism (KEM). KEM takes as input a public key and produces a random symmetric key of a pre-specified length and an encryption of that key.

A. System Architecture

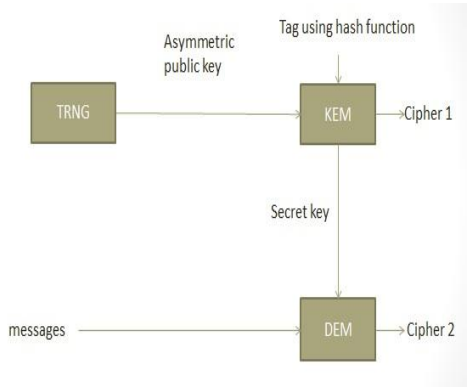

Fig 1: PROPOSED ARCHITECTURE

- In this system TRNG based on Mouse movement generates the Asymmetric public key which is used to generate the secret key via KEM.
- KEM splitting the key space into two parts , one is a secret key which is used to encrypt original messages in the DEM .another one is the Encryption data which is used to verify the message's non reputation in receiver side.
- KEM take the pair of Mouse co-ordinates as a Inputs and generate the public key and splitting the keys set into two parts
- Cipher 1, which is used to generate the data encryption key in receiver side.
- Data encryption key (k), which is used to encrypt the original messages in DEM.

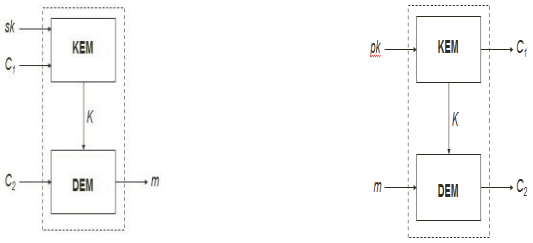

Fig 2.a: encryption of message

Fig 2.b: decryption of message

The proposed work contains the following modules

Node Deployment Based on Sink

To implement the Project concept, first we have to construct a sink which consists of 'n' number of Nodes. Sink is used to store node id and other information. So that nodes can request data from other nodes in the sink. All nodes are connected in to sink. Also the sink will monitor all the Nodes Communication for security purpose.

Random number and Master Pair generation

In this module we focus on the true random number generation from the user's mouse movement pattern.

The activities involved with this module

- Gathering the X, Y co-ordinates from the user mouse movement pattern.
- Generating the public key from the random numbers.

In this module sink generates a master pair key for each node from mouse patterns and node ID. Master pair key includes public key and private key. Using MD5 algorithm for hashing the node id to generates the master pair key. If the source node wants to send data to destination node means then only source node gives a request to sink for knowing key to encrypt the data. Then only sink gives a master pair key to source node. So it provides high security for data transfer.

Key encapsulation

In this module we focus on the private key generation for the data encapsulation module using KEM mechanism. Public key which is generated in the previous phase provided as a Input and it spited into two parts: cipher data which is used to retrieve the secret key in another side, and symmetric key which is used to encrypt the original data in data encryption module. In this module, source node request to sink for generating symmetric key using signcryption algorithm after master pair key generation. Sink concates node id with private key (master pair key).

So It produces the partial secret key. Although sink concates this partial secret key with public key (master pair key). It produces the user generated pair key. Then it splits this key to two, that is, first part is like a public key and second part is like a private key.

Encryption based on Symmetric Key

After creation of final public key and private key, then source node request to sink for encrypt the original message. Sink concates source node id, public key and private key, destination node id and destination node

public key (master pair key). It produces the symmetric key.

Then the sink encrypts the original message using symmetric key. After encryption it creates the cipher text. Finally cipher text transfer to source node using sink as a response. Then source node id, destination node id with cipher text goes to destination node via intermediate nodes.

Data Transmission and Decryption

In this module data reaches the destination via intermediate nodes. Then destination node already knows the master pair key using sink. Destination requests to sink to decrypt the cipher text. So sink concates creates partial key. Using partial key with its public key generates user generated pair key. Then sink creates public key and private key by splitting its user generated pair key. Finally it creates symmetric key using destination node id, public key, private key, source node id and public key (source node public key). Then it decrypts the cipher text using symmetric key. Finally it displays the original data to destination node.

V. Implementation and Results

We were generating the simulation codes and check the performance of the methodology in the manner of security and randomness. In this work we focus on the statistical experiments about the randomness and security of the TRNG which is proposed and compare with the Pseudo random number generators.

The simulations were carried out with 25,50and 100 nodes topologies, where 10 nodes in each of these topologies were designated to send requests simultaneously to BS. It was found that proposed key generation outperforms all the existing methods we considered in the study with respect to the parameters latency and security.

Latency means the time taken for a Request to travel from Requester node to BS. In this work we considered three requests Key_gen_req, Signcrypt_req and Designcrypt_req.

The simulation experiments were carried out in NS-2.32 (http://www.isi.edu/nsnam/ns/). The Network settings for the Simulation experiments are as follows.

(a) Terrain dimensions: 2000 m _ 2000 m
(b) Simulation time: 15, 30 and 60 min
(c) Number of nodes: 50, 100, 500 and 1000
(d) Mobility model: Random Way Point
(e) Speed of the mobile nodes: 0
(f) Underlying MAC protocol: IEEE 802.11.
(g) Channel: /Wireless
(h) Propagation: TwoRayGround
(i) Network type: WirelessPhy
(j) Queue: Drop Tail/PriQueue
(k) Antenna: Omni Antenna
(l) Topography 700; # X dimension of the topography 700; # Y dimension of the topography

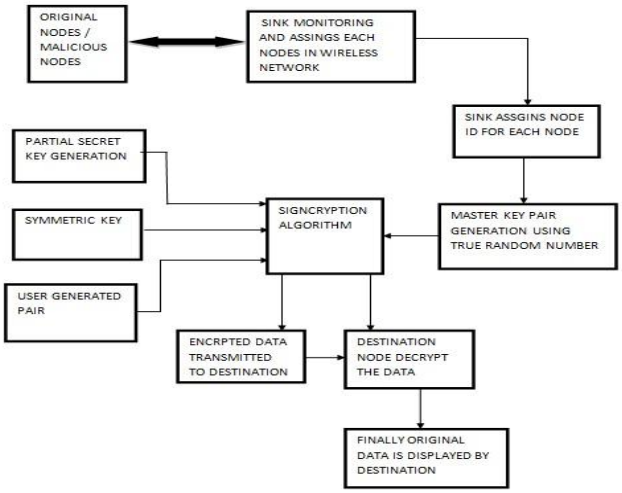

Fig 3: Simulation model

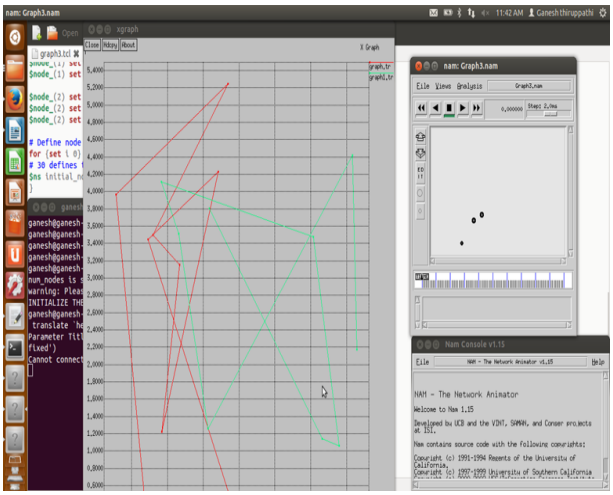

Fig 4: Simulation snapshot

First we generate Random master key pairs from User's mouse movement patterns. We draw graphs using values of the random numbers which is generated from mouse value and store it in BS. Every 250 micro sec mouse patterns value should be updated and graph will be change.so it will provide high security for key generation even the insider should not know about the patterns of the pair generation. Then Nodes which are interested to participate in communication sent the key_gen_req for got a new master pair for message communication.

When BS receives the key generation requests from nodes it provides a unique and unpredictable master pair for the communication and store that value in the table mentioned by destination node id. In the reply node got the master pair then it generates user key pairs (user public key, user private key) and also generates secret key values for signcryption. It sends the request signcrypt_req to BS with following values source ID, destination ID, message, source public & private key and destination public key.

BS generates the symmetric key and cipher of that key and done the encryption of message also. It returns the cipher as reply to source .for the Unsigncryption at the destination end it will send designcrypt_req to BS with the following attributes. It returns the original messages to the destination.

VI. CONCLUSION

In this paper, we design a secure TKEM, with the help of true random number generators. In this method true random numbers are generated using user mouse movement patterns, which is unpredictable and non-deterministic to the attacker to crack the secure code. Thus this method we can create the true random numbers without any complex external equipment. Normal computer mouse plays an important role Extensive analyses show that our proposed scheme satisfies the desired security requirements and guarantees efficiency as well.

REFERENCES

[1]. Yuliang Zheng,"Signcryption and its applications in efficient public key solutions" Information Security Computer Science Volume 1396, 1998, pp 291-312.

[2]. Dr Mark D. Ryan: Public Key Encryption, Lecture Notes, University of Birmingham (2004).

[3]. Fagen Li, Masaaki Shirase, and Tsuyoshi Takagi, "Certificate less Hybrid Signcryption", 2009.

[4]. Shamir, "Identity-based cryptosystems and signature schemes", Advances in Cryptology, volume 196 of LNCS 1985.

[5]. Al-Riyami and Paterson, "Certificate less public key cryptography", Advances in Cryptology-ASIACRYPT, volume 2894, 2003.

[6]. M. Barbosa, P. Farshim, "Certificateless signcryption", In: ACM Symposium on Information, Computer and Communications Security-ASIACCS 2008, pp. 369–372.

[7]. S. Sharmila Deva Selvi, S. Sree Vivek and C. Pandu Rangan, "Certificate less KEM and Hybrid Signcryption Schemes Revisited", 2006.

[8]. Wenjian Xie and Zhang Zhang, "Certificate less Signcryption without Pairing", 2009.

[9]. P.Murali and R.Palraj, "True Random number generator method based on image for key exchange algorithm", Vol.1, 2011.

[10]. Mohsen Toorani, Ali A.Beheshti, "An Elliptic Curve-based Signcryption Scheme with Forward Secrecy", Journal of Applied Sciences, Vol. 9, 2009.

[11]. Malone-Lee, "Identity-Based Signcryption" Cryptology ePrint Archive, Report 2002/098. 2002.