April - June © 2015 Transactions; 2(6): 263-274

International Journal of Research and Applications

ISSN (online): 2349-0020 ISSN (print): 2394-4544 http://www.ijraonline.com/

Survey Report

A Survey on new approaches in SignCryption

Varun Kumar Chennuri [1] and Dr. Sadanandam Manchala (Supervisor) [2]

Corresponding Author:
kumarvarun501@gmail.com

DOI:
http://dx.doi.org/
10.17812/IJRA.2.6(48)2015

Manuscript:
Received: 2nd April, 2015
Accepted: 15th May, 2015
Published: 12th June, 2015

Publisher:
Global Science Publishing Group, USA
http://www.globalsciencepg.org/

ABSTRACT

SignCryption is a new-fangledmodel in public key type ofcryptography to provide confidentiality and authentication in a solitary logical step at the lower computation cost and communiqué overhead compared to the long-standing signature-then-encryption mechanism [1]. This paper may contain few assessment documents exist, which described that how long had been the signcryption schemes used by the authors. We propose a new style to implementing existing signcryption with the new scheme. Our offered scheme is aprogressed version of existing mechanism followed in Bao & Deng, in which publicly verifiable signcryption is designed [3]. In this scheme, we make sure the security feature of forward secrecy to the signcryption, without an increase in computational cost. Also, this new signcryption procedure delivers the security services of message confidentiality and Authentication using public verifiability.

Keywords: SignCryption, Public Key Cryptography, Signature-then-Encryption, Public Verifiability, Forward Secrecy.

[1 2] Department of Computer Science and Engineering, KUCE & T
[1 2] Kakatiya University, Warangal, Telangana, India.

IJRA - Year of 2015 Transactions:
Month: April - June
Volume – 2, Issue – 6, Page No's:263-274
Subject Stream: Computers

Paper Communication: Author Direct

Paper Reference Id: IJRA-2015: 2(6)263-274

DOI: 10.17812/IJRA.2.6(48)2015

International Journal of Research and Applications
Apr-Jun © 2015 Transactions 2(6): 263-274

eISSN : 2349 - 0020
pISSN : 2394 - 4544

COMPUTERS SURVEY REPORT

A Survey on new approaches in SignCryption

Varun Kumar Chennuri [1] and Dr. Sadanandam Manchala (Supervisor) [2]
[1 2] Department of Computer Science and Engineering, KUCE & T,
[1 2] Kakatiya University, Warangal, Telangana, India.
[1] kumarvarun501@gmail.com, [2] sadanb4u@gmail.com

ABSTRACT

SignCryption is a new-fangledmodel in public key type ofcryptography to provide confidentiality and authentication in a solitary logical step at the lower computation cost and communiqué overhead compared to the long-standing signature-then-encryption mechanism [1]. This paper may contain few assessment documents exist, which described that how long had been the signcryption schemes used by the authors. We propose a new style to implementing existing signcryption with the new scheme. Our offered scheme is aprogressed version of existing mechanism followed in Bao & Deng, in which publicly verifiable signcryption is designed [3]. In this scheme, we make sure the security feature of forward secrecy to the signcryption, without an increase in computational cost. Also, this new signcryption procedure delivers the security services of message confidentiality and Authentication using public verifiability

Keywords: SignCryption, Public Key Cryptography, Signature-then-Encryption, Public Verifiability, Forward Secrecy.

I. INTRODUCTION

The encryption and digital signature are the two basic cryptographic mechanisms that can provide the security of communications. Until the decade before, they have been viewed as important but distinct building blocks of various cryptographic systems. Cryptography arose as a means to enable parties to maintain the privacy of the information they send to each other, even in the presence of an adversary with access to the communication channel. While providing privacy remains a central goal, the field has expanded to encompass many others, including not just other goals of communication security, such as guaranteeing integrity and authenticity of communications, but many more sophisticated and fascinating goals. Once largely the domain of the military, cryptography is now in widespread use, and you are likely to have used it even if you don't know it. When you shop on the Internet, for example to buy a book at www.amazon.com, cryptography is used to ensure privacy of your credit card number as it travels from you to the shop's server. Or, in electronic banking, cryptography is used to ensure that your checks cannot be forged. Cryptography has been used almost since writing was invented. For the larger part of its history, cryptography remained an art, a game of ad hoc designs and attacks [10]. Although the field retains some of this flavor, the last twenty-five years have brought in something new. The art of cryptography has now been supplemented with a legitimate science. In this course we shall focus on that science, which is modern cryptography. Modern cryptography is a remarkable discipline. It is a cornerstone of computer and communications security, with end products that are imminently practical.

In the public key schemes, a traditional method is to digitally sign a message, then followed by an encryption (signature-then-encryption) that can have two problems: Low efficiency and high cost of such summation, and the case that any arbitrary scheme cannot guarantee the security. The signcryption is a relatively new cryptographic technique that is supposed to fulfill the functionalities of digital signature and encryption in a single logical step. It effectively decreases the computational costs and communication overheads in comparison with the traditional Signature-then-encryption schemes. The first signcryption scheme was introduced by Zheng (1997) but it fails the *forward secrecy* of message confidentiality and verifying of a signature not in publicly [2]. Several signcryption schemes have also proposed over the years, each of them providing different levels of security services and computational costs. To overcome this *Bao&Deng* proposed new Signcryption scheme which is modified version of Zheng scheme. In this scheme public verifiability is available.

Applications of Signcryption

A major motivation of signcryption is to search for a more economical method for secure and authenticated transactions/message delivery. The proposed signcryption schemes are compact and particularly suitable for smart card based applications. We envisage that they will end innovative applications in many areas including digital cash payment systems, EDI and personal heath cards [10]. An important fact is that signcryption can be used to design more efficient digital cash transaction protocols that are often required to provide with both the functionality of digital signature and encryption.

A signcryption scheme should produce a signcryption "ciphertext" which is shorter than a naive combination of apublic-key encryption ciphertext and a digital signature.

A signcryption scheme should provide greater security guarantees and/or greater functionality than a native combination of public-key encryption and digital signatures [1]. More recently, the significance of signcryption in real-world applications has gained recognition by experts in data security. Since 2007, a technical committee within the International Organization for Standardization (ISO/IEC JTC 1/SC 27) has been developing an international standard for signcryption techniques [7]. The shared secret key between the parties makes possible an unlimited number of applications. Among these applications, one can first think of the following three:

- Secure and authenticated key establishment,
- Secure multicasting, and
- Authenticated key recovery.

A number of signcryption-based security protocols have been proposed for aforementioned networks and similar environments the mobile ad hoc networks get subjected to security threats like other wireless networks. But due to their peer to peer approach and the absence of infrastructural resources the mobile adhoc networks cannot use strong cryptographic mechanisms as used by their other wireless counterparts. This led to the development of trust based methods as security solutions wherein a trusted node is relaxed from security checks when the trust value reaches to a particular limit. The trust methods are prone to security risks, but have found their acceptance due to efficiency over computationally expensive and time consuming cryptographic methods. The major problem with the trust methods is the period during which trust is growing and is yet to reach the requisite threshold. There are also various applications of signcryption in electronic commerce, where its security properties are very useful. Analyzing this security scheme from an application-oriented point of view [10], can be observed that a great amount of electronic commerce can take advantage of signcryption to provide efficient security solutions in the following areas:

- Electronic payment,
- Electronic toll collection system,
- Authenticated and secured transactions with smart cards, etc.

Public key cryptography

Public key cryptography is sometimes also referred to as asymmetric cryptography. Public key cryptography is a relatively new field, invented in 1975 [DIFF76b] (at least that's the first published record-it is rumored that the NSA or similar organizations may have discovered this technology earlier). Unlike secret key cryptography, keys are not shared. Instead, each individual has two keys: a private key that need not be revealed to anyone, and a public key that is preferably known to the entire world. Note that we call the private key a *private key* and not a *secret key [5]*. This convention is an attempt to make it clear in any context, whether public key cryptography or secret key cryptography is being used. There are people in this world whose sole purpose in life is to try to confuse people. They will use the term *secret key* for the private key in public key cryptography, or use the term *private key* for the secret key in secret key technology. One of the most important contributions we can make to the field is to convince people to feel strongly about using the terminology correctly-the term *secret key* refers only to the single secret number used in secret key cryptography. The term *private key* must be used when referring to the key in public key cryptography that must not be made public. (Yes, when we speak, we sometimes accidentally say the wrong thing, at least we feel guilty about it.) There is something unfortunate about the terminology *public* and *private*. It is that both words begin with *p*. We will sometimes want a single letter to refer to one of the keys. The letter *p* won't do. We will use the letter *e* to refer to the public key, since the public key is used when encrypting a message. We'll use the letter *d* to refer to the private key, because the private key is used to decrypt a message. Encryption and decryption are two mathematical functions that are inverses of each other. In doing the two-step approach has been followed. Namely, before a message is sent out, the sender of the message would sign it using a digital signature scheme, and then encrypts 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 two-step approach signature-then-encryption.

Signature generation and encryption consume machine cycles, and also introduce expanded" bits in an original message. Symmetrically, a comparable amount of computation time is generally required for signature verification and decryption [7]. 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 current standard signature-then-encryption approach, the cost of delivering a message in a secure and authenticated way is essentially the sum of the cost for digital signature and that for encryption.

The symmetric setting

The simplest and also most common setting is that the sender and receiver share a *key* that the adversary does not know. This is called the *symmetric setting* or symmetric trust model. The encapsulation and decapsulation procedures would both depend on this same shared key. The shared key is usually a uniformly distributed random string having some number of bits, *k*. Recall that a *string* is just a sequence of bits. The sender and receiver must somehow use the key *K* to overcome the presence of the adversary. One might ask how the symmetric setting is realized. The symmetric model is not concerned with how the parties got the key, but with how to use it. In cryptography we assume that the secret key is kept securely by the party using it. If it is kept on a computer, we assume that the adversary cannot penetrate these machines and recover the key. Ensuring that this assumption is true is the domain of computer systems security.

Let us now take a closer look at some specific problems in the symmetric setting. We will describe these problems quite informally, but we will be returning to them later in our studies, when they will get a much more thorough treatment.

Symmetric Encryption Schemes: A protocol used to provide privacy in the symmetric setting is called a *symmetric encryption scheme*. When we specify such a scheme Π, we must specify three algorithms, so that the scheme is a triple of algorithms, Π = (*K*, *E*, *D*). The encapsulation algorithm we discussed above is, in this context, called an *encryption* algorithm, and is the algorithm *E*. The message *M* that the sender wishes to transmit is usually referred to as a *plain text*. The sender *encrypts* the plaintext under the shared key *K* by applying *E* to *K* and *M* to obtain a *cipher text C*. The cipher text is transmitted to the receiver. The above-mentioned decapsulation procedure, in this context, is called a *decryption* algorithm, and is the algorithm *D*. The receiver applies *D* to *K* and *C*. The decryption process might be unsuccessful, indicated by its returning a special symbol ⊥, but, if successful, it ought to return the message that was originally encrypted. The first algorithm in Π is the *key generation* algorithm which specifies the manner in which the key is to be chosen. In most cases this algorithm simply returns a random string of length the key length. The encryption algorithm *E* may be randomized, or it might keep some state around The encryption scheme does not tell the adversary what to do. It does not say how the key, once generated, winds its way into the hands of the two parties. And it does not say how messages are transmitted. It only says how keys are generated and how the data is processed.

Message Authenticity: In the message-authentication problem the receiver gets same message which is claimed to have originated with a particular sender. The channel on which this message flows is insecure. Thus the receiver *R* wants to distinguish the case in which the message really did originate with the claimed sender *S* from the case in which the message originated with some imposter, *A*. In such a case we consider the design of an encapsulation mechanism with the property that un-authentic transmissions lead to the decapsulation algorithm outputting the special symbol ⊥. The most common tool for solving the message-authentication problem in the symmetric setting is a *message authentication scheme*, also called a *message authentication code* (MAC)[10]. Such a scheme is specified by a triple of algorithms, Π = (*K*, *T*, *V*). When the sender wants to send a

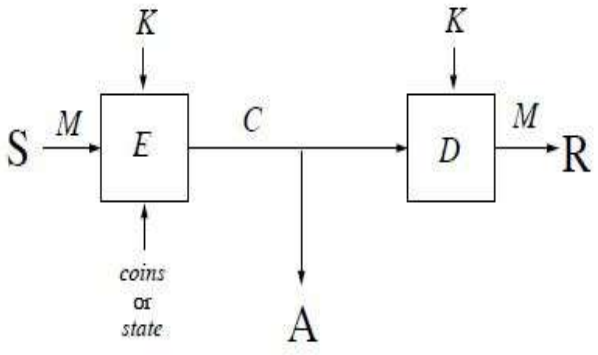

Figure 1.3: Symmetric encryption. The sender and the receiver share a secret key, *K*. The adversary lacks this key. The message *M* is the plaintext; the message *C* is the cipher text.

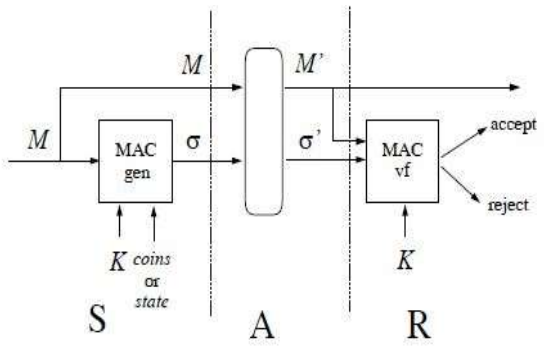

Figure 1.4: A message authentication code. The tag σ accompanies the message *M*. The receiver *R* uses it to decide if the message really did originate with the sender *S* with whom he shares the key *K*. Message *M* to the receiver she computes a "tag," σ, by applying *T* to the shared key *K* and the message *M*, and then transmits the pair (*M*, σ). (The encapsulation procedure referred to above thus consists of taking *M* and returning this pair. The tag is also called a MAC.) The computation of the MAC might be probabilistic or use state, just as with encryption. Or it may well be deterministic. The receiver, on receipt of *M* and σ, uses the key *K* to check if the tag is OK by applying the *verification algorithm V* to *K*, *M* and σ. If this algorithm returns 1, he accepts *M* as authentic;

otherwise, he regards *M* as a forgery. An appropriate reaction might range from ignoring the bogus message to tearing down the connection to alerting a responsible party about the possible mischief. See Figure 1.4.

The asymmetric setting

A shared key *K* between the sender and the receiver is not the only way to create the information asymmetry that we need between the parties and the adversary. In the *asymmetric setting*, also called the *public-key setting*, a party possesses a *pair* of keys-a *public key*, *pk*, and an associated *secret key*, *sk*. A party's public key is made publicly known and bound to its identity. For example, a party's public key might be published in a phone book.

The problems that arise are the same as before, but the difference in the setting leads to the

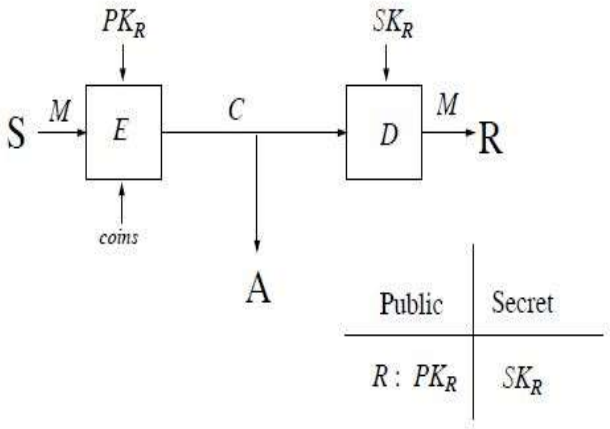

Figure 1.5: Asymmetric encryption. The receiver *R* has a public key, *pkR*, which the sender knows belongs to *R*. The receiver also has a corresponding secret key, *skR*. Development of different kinds of tools.

Asymmetric Encryption: The sender is assumed to be able to obtain an authentic copy *pkR* of the receiver's public key. (The adversary is assumed to know *pkR* too.) To send a secret message *M* to the receiver the sender computes a cipher text $C \leftarrow EpkR(M)$ and sends *C* to the receiver. When the receiver receives a cipher text *C* he computes $M \leftarrow DskR(C)$. The asymmetric encryption scheme $\Pi = (K, E, D)$ is specified by the algorithms for key generation, encryption and decryption. For a picture of encryption in the public-key setting, see Fig. 1.5. The idea of public-key cryptography, and the fact that we can actually realize this goal, is remarkable. You've never met the receiver before. But you can send him a secret message by looking up some information in a phone book and then using this information to help you garble up the message you want to send. The intended receiver will be able to understand the content of your message, but nobody else will. The idea of public-key cryptography is due to Whitfield Diffie and Martin Hellman and was published in 1976.

Digital Signatures: The tool for solving the message-authentication problem in the asymmetric setting is a *digital signature*. Here the sender has a public key *pkS* and a corresponding secret key *skS*. The receiver is assumed to know the key *pkS* and that it belongs to party *S*. (The adversary is assumed to know *pkS* too.) When the sender wants to send a message *M* she attaches to it some extra bits, σ, which is called a *signature* for the message and is computed as a function of *M* and *skS* by applying to them a *signing* algorithm Sign. The receiver, on receipt of *M* and σ, checks if it is OK using the public key of the sender, *pkS*, by applying a *verification* algorithm *V*. If this algorithm accepts, the receiver regards *M* as authentic; otherwise, he regards *M* as an attempted forgery. The digital signature scheme $\Pi = (K, \text{Sign}, V)$ is specified by the algorithms for key generation, signing and verifying. A picture is given in Fig. 1.6.

One difference between a MAC and a digital signature concerns what is called *non-repudiation*. With a MAC anyone who can verify a tagged message can also produce one, and so a tagged message would seem to be of little use in proving authenticity in a court of law. But with a digitally-signed message the *only* party who should be able to

produce a message that verifies, under public key *pkS* is the party *S* herself. Thus, if the signature scheme is good, party *S* cannot just maintain that the receiver, or the one presenting the evidence, concocted it. If signature σ authenticates *M*

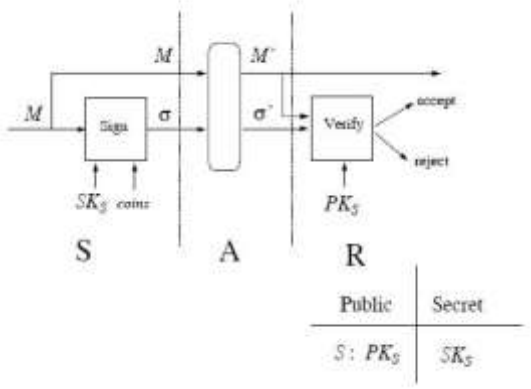

Figure 1.6: A digital signature scheme. The signature σ accompanies the message *M*. The receiver *R* uses it to decide if the message really did originate with the sender *S* with has public key *pkS*.

Public verifiability

Normally, in a signcryption scheme, the message is hidden and thus the validity of the ciphertext can be verified only after unsigncrypting the ciphertext. Thus, a third party who is unaware of the receiver's private key will not be able to verify whether a ciphertext is valid or not. Public verifiable signcryption schemes are applicable in filtering out the spams in a secure email system [7]. The spam filter should be able to verify the authenticity of the ciphertext without knowing the message (i.e., check whether the signcryption is generated from the claimed sender or not). Moreover, in applications such as private contract signing, made between two parties, the receiver of the signcryption should be able to convince the third party that indeed the sender has signed the corresponding message hidden in the signcryption. In this case, the receiver should not reveal his secret key in order to convince the third party, instead he reveals the message and some component computable with his private key required for the signature verification. In literature, signcryption schemes in which a third party can verify the validity of the cipher text without the knowledge of the hidden message, or without knowing the receiver private key are called third party verifiable signcryption schemes. To the best of our knowledge, Bao[3] proposed the first public verifiable signcryption scheme in the PKI based setting. Following that, a number of schemes [5] were proposed in the PKI based setting. Chang [11] proposed an identity based signcryption scheme that provides both public verifiability and forward security. To the best of our knowledge the scheme in is the only identity based scheme providing public verifiability and third party verification.

Forward Secrecy of message confidentiality:

The security of communications transmitted across the Internet can be improved by using public key cryptography. However, if the public and private keys used in those communications are compromised, it can reveal the data exchanged in that session as well as the data exchanged in previous sessions. The concept of Forward Secrecy (FS) is the property that ensures that a session key derived from a set of long-term public and private keys will not be compromised if one of the (long-term) private keys are compromised in the future [11]. Online systems such as IPSEC can negotiate new keys for every communication and if a key is compromised only the specific session it protected will be revealed. For Forward Secrecy to exist the key used to protect transmission of data must not be used to derive any additional keys, and if the key used to protect transmission of data was derived from some other keying material, that material must not be used to derive any more keys.It means that even if the long-term private key of the sender is revealed, the adversary is not capable of decrypting the previously signcrypted texts. The only way to defeat forward secrecy is that the adversary should possess any other secret information of sender apart from his /her private key. In most schemes this other secret corresponds to random number or hashed value.

II. LITERATURE SURVEY

1. Traditional Signature-then-Encryption

Public key cryptography developed by Diffe and Hellman makes it a reality for one to digitally sign a message, and another to send a message securely to another person with whom no common encryption key has been shared [1]. Some of the most important public key digital signature/encryption schemes, these being RSA encryption and signature scheme, ElGamal encryption and signature scheme, and two signature schemes derived from ElGamal, namely Schnorr signature scheme and Digital Signature Standard (DSS).

Message Encryption

Encryption means conversion of messages from a comprehensible form into an incomprehensible one and back again at the other end, rendering it unreadable by interceptors or eavesdroppers without secret knowledge (namely the key needed for decryption of that message). The sequence of data processing steps required for the transformation of the plaintext into cipher text is called message encryption. Various parameters used by an encryption algorithm, are derived from a secret key. As discussed in the previous Chapter we have a number of encryption algorithms. *DES* or *AES* can be used for message encryption.

Digital Signature

In the digital signature process *Alice* is the sender and *Bob* is the receiver. *Alice* uses a signing algorithm to sign the message. The message and the signature are sent to the receiver. Then the receiver receives both and applies the verifying algorithm whether to accept the message or not. Several digital Signatures Schemes have been evolved during the last few decades. Some of them have been implemented. They are: *RSA Digital Signature Scheme*, *ElGamal Digital Signature Scheme*, *Schnorr Digital Signature Scheme*, *Digital Signature standard (DSS)*, *Elliptic Curve Digital signature Scheme*. Now we can apply both the operations one after other to provide message confidentiality and authenticity. This is known as *Signature-Then-Encryption"*.

Signature-then-Encryption

In order to send a confidential letter in a way that it can't be forged, it has been a common practice for the sender of the letter to be sign it, put it in an envelope and then seal it before handing it over to be delivered [1]. Discovering public key cryptography has made communication between people who have never met before over an open and insecure network such as Internet in a secure and authenticated way possible. Before sending a message the sender has to do the following:

1. Sign it using a digital signature scheme (DSS)
2. Encrypt the message and the signature using a private key encryption algorithm under randomly chosen encryption key.
3. Encrypt the random message encryption key using receiver's public key. This approach is known as *Signature-Then-Encryption*

2. Zeng's SignCryption

Public key cryptography discovered nearly two decades ago [5] has revolutionized the way for people to conduct secure and authenticated communications. It is now 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 the Internet. In doing so the same two-step approach has been followed. Namely, before a message is sent out, the sender of the message would sign it using a digital signature scheme, and then encrypts the message (and the signature) using 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 two-step approach signature-then-encryption Signature generation and encryption consume machine cycles, and also introduce expanded" bits in an original message. Symmetrically, 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 current standard signature-then-encryption approach, the cost of delivering a message in a secure and authenticated way is essentially the sum of the cost for digital signature and that for encryption.

The cost of secure and authenticated message delivery, namely, whether 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. This question seems to have never been addressed in the literature since the invention of public key cryptography. Finally he discovers a new cryptographic primitive termed as signcryption" which simultaneously fulfill both the functions of digital signature and public key encryption in a logically single step [2], and with a cost significantly smaller than that required by signature-then-encryption.

Any SignCryption scheme should have the following properties:

1. Correctness: There exist an unSignCryption schemes from which the plain text can be recovered from the signcrypted message.

2. Efficiency: A SignCryption scheme is said to be efficient if the computational cost and the communication cost should be smaller than that of signature-than-encryption standard.

3. Security: It should fulfill the security properties of both digital signature and encryption standard. Some of the security issues are discussed hereunder:

- ***Confidentiality:*** It should be infeasible for an eavesdropper to get any information from the signcrypted message without knowing the sender's and receiver's private key.
- ***Integrity:*** The intended or authenticated user can only modify the content of the message.
- ***Unforgeability:*** There should not be two signcrypted messages which give the same plaintext. Otherwise an adaptive attacker can create an authentic signcrypted text that can be accepted by the unSignCryption algorithm.
- ***Forward Secrecy*** If the long term private key of the sender is compromised, no one should be able to extract any information of the past messages.
- ***Non-repudiation*** after sending the message later Alice should not deny that she has sent the message or after receiving the message Bob cannot deny that he has received the message.
- Public Verifiability Any third party or judge can verify whether the message has been sent by the intended user

Implementation Work on ZENG Scheme:

Alice has a message m to send to Bob. Alice sign crypts m so that the effect is similar to the signature-then-encryption.

Public Parameters

The public parameters used in the process of SignCryption and unSignCryption are given below:

- p — a large prime.
- q — a large prime factor of p-1.
- g — an integer with order q modulo p chosen randomly from [1,...,p-1].
- Hash — a one way hash functions.
- KH — a keyed one way hash functions.
- Ek(.)/*Dk* (.) — Symmetric encryption/decryption algorithm with private key *k* such as *DES* or *AES*.

***Alice's keys*:**

1. *xa* — *Alice's* private key chosen at random from [1,...,q-1].

2. *ya* — *Alice's* public key $ya = g^{xa} \bmod p$.

Bob's keys:

1. *xb* — Bob's private key chosen at random from [1,...,q-1].

2. *yb* —Bob's public key ($yb = g^{xb} \bmod p$)

Signcrypting:

In this *Alice*, sends the signcrypted message to the recipient *Bob*. First she digitally signs the message then encrypts it and sends it to *Bob*.

SignCryption of a message by Alice the sender

• Choose a number x at random from the set [1... q-1]. And compute k = hash ($y_b^x \bmod p$).

Split k into *k*1 and *k*2 of equal length.

• Calculate $r = KH_{k2}(m)$.

• c = $E_{k1}(m)$.

•$s = x = (r + x_a) \bmod q$ if SDSS1 is used. Or

•$s = x = (1 + x_a . r) \bmod q$ if SDSS2 is used instead.

• Send (c, r, s) to *Bob* the recipient.

Unsigncrypting:

In this Bob decrypts the message sent by Alice and verifies the authenticity of the message

•Compute k from r, s, g, p, *ya*, *xb*.

$k = hash((ya . gr)^{s.xb} \bmod p)$ if SDSS1 is used, or

$k = hash((g . y^r a)^{s.xb} \bmod p)$ if SDSS2 is used

•Split k into *k*1 and *k*2 of equal length.

• Calculate m = *Dk*1(*c*).

Accept m if $KH_{k2}(m) = r$. It ensures that the message has come from Alice. Otherwise he rejects.

Thre two singcryption schemes were given, called SDSS1 and SDSS2. Here we only describe the case for SDSS1. The case for SDSS2 is similar [5]. In Zheng's unsigncrypting process, it is straightforward to see that x_b is involved for signature verification. Hence in scheme public verifiability is not possible.

3. Deng's SignCryption

In Zheng scheme, receiver's private key is no longer needed in verifying signature. In Bao&Deng Signcryption, signature is directly verifiable by sender's public key [3]. But the computational cost of the Deng scheme is higher than that of Zeng' scheme, but lower than that of signature-then-encryption approach. The correctness, efficiency, and security are the essential attributes that any signcryption scheme should take them into account. A signcryption scheme should simultaneously fulfill the security attributes of an encryption and those of a digital signature.Such security services mainly include: *Confidentiality, Unforgeability*

Implementation Work of BAO & DENG Scheme:

Public Parameters

- P - a large prime number
- Q - a large prime factor of p-1
- G - an integer with order q modulo p chosen randomly from [1,…, p-1]
- Hash - a one-way hash function whose output has, say, at least 128 bits
- KH - a keyed one-way hash function
- (E, D) - the encryption and decryption algorithms of a private key cipher (Any symmetric key Algorithms like DES, 3DES, AES, etc.).

Alice's keys

- Xa - Alice's private key, chosen uniformly at random from [1… q-1]
- Yo - Alice's public key ($y_a = g^{xa} \bmod p$)

Bob's keys

- Xb - Bob's private key, chosen uniformly at random from [1… q-1]
- Yb - Bob's public key ($Yb = g^{xb} \bmod p$)

Signcryption at Sender:

- In order to signcrypt a message m to Bob, Alice has to accomplish the following operations:

  Choose a random number x€ R Z^*_q then sets
- Calculate $t1 = g^x \bmod p$
- Calculate $t2 = (Yb)^x \bmod p$
- Calculate $c = E_{hash\ (t2)}$ (m)
- Calculate r = hash (m, t_1)
- Calculates = x /(r +X_a) mod q
- Alice sends to Bob the values (c, r, s).

Unsigncryption at receiver:

- In order to unsigncrypt a message from Alice, Bob has to accomplish the following operations:
- Calculate k using r, s, g, p, y_a and x_b
- Calculate $t_1 = (y_a g^r)^s \bmod p$
- Calculate $t_2 = (t_1)^{xb} \bmod p$
- Calculate $m= D_{hash\ (t2)}$ (c)
- Check whether r=hash (m, t_1)

  Bob may pass (c, r, s) to others, who can be convinced that it indeed came from Alice by Verifying

  $$r=\text{hash}\ (m,(y_a g^r)^s)$$

Drawbacks of Bao&Deng Scheme Even at an increase computational cost, Bao&Deng scheme does not provide the security feature of Forward Secrecy.

4. Zheng-Imai Elliptic Curve Signcryption Scheme

The Two most popular schemes named as ECSCS1 and ECSCS2 based on elliptic curved are purposed by Zheng–Imai. We are discussing only ECSCS1. The case is similar for the other ECSCS2. If Alice wants to send a message m to Bob he has to signcrypts m as follows [8].So that the effect was similar to signature then encryption.

Public Parameters:

•C — Consider C as an elliptic curve over a finite field GF(p^m), either with p ¸2^{160} and m=1 or p=2 and m¸160.

• q — a large prime whose size is approximately of orderp^m-1.

•G — a point with order q. Chosen randomly from thepoints on C.

• hash (.) — a one way hash function whose output has sayat least 160 bits.

• KH (.)— a keyed one-way hash function.

• (E, D) — the encryption and decryption algorithms of a private key cipher.

Alice's keys:

• v_a| Alice's private key chosen uniformly at random from [1,...,q-1].

• P_a| Alice's public key. ($Pa = v_a G$, a point on C).

Bob's keys:

• v_b | Bob's private key chosen uniformly at random from [1,...,q-1].

• P_b | Bob's public key. (P_b= v_bG , a point on C).

SignCryption scheme by Zheng and Imai

V ² [1,...,q-1].A random number chosen by Alice.

($k1$; $K2$)=hash (vPb).

c=E_{k1} (m).

r=KH_{k2} (m; *blind info*).

s= $v/(r+v_a)$ mod q.

Send c,r,s to Bob.

UnSignCryption scheme by Zheng and Imai

u=s$_v$$b$ mod q.

($k1$; $K2$)=hash (uPa + urG).if SECDSS1 is used, or

($k1$; $K2$)=hash (uG + urPa).if SECDSS2 is used.

m=$Dk1$(c).

Accept m only if KH_{k2} (m; *blind info*) =r.

The disadvantage of the above scheme is that it doesn't support forward secrecy and encrypted message authentication. From the above Zheng and Imai scheme we can see that if *Alice* divulged his private key *va* inattentively then an adversary can get the information about the past messages. Now let's discuss Hwang et al. SignCryption scheme based on elliptic curve cryptosystem, which provides forward secrecy.

5. Schnorr Signcryption

A Schnorr signature is a digital signature produced by the Schnorr signature algorithm. Its security is based on the intractability of certain discrete logarithm problems. It is considered the simplest digital signature scheme to be provably secure in a random oracle model [4].

Choosing parameters

All users of the signature scheme agree on a group G with generator g of prime order q in which the discrete log problem is hard.

Key generation

Choose a private signing key x. The public verification key is $y = g^x$.

SignCryption at sender

To sign a message M: , Choose a random k.

Let $r = g^k$

Let e = H(M | | r), where || denotes concatenation and r is represented as a bit string. H is a cryptographic hash function

Let s = (k − xe). The signature is the pair (s,e).

UnSinCryption at receiver

Let $r_v = g^s y^e$

Let e_v = H(M | | r_v)

If e_v = e then the signature is verified.

III.CONCLUSION

In this paper, a new Signcryption scheme is introduced that simultaneously provides the security attributes of message confidentiality, authentication, Integrity, unforgetability, and non-repudiation. It also provides the security attribute of public verifiability, so that any trusted third party can verify the sender's signature. But this paper, describes various Signcryption scheme papers are involved and that can be implementation should be interleaved in it. Because our proposed scheme is implemented from exsitng wholly papers. We amend existing Bao & Deng scheme so that our scheme provides more security feature of Forward Secrecy in addition to the features provided by the Bao&Deng Scheme in same computational cost as of the existing scheme

IV. REFERENCES

[1] MihirBellare1, PhillipRogaway2, "Introduction to Modern Cryptography",Department of Computer Science and Engineering, University of California at San Diego, La Jolla,CA92093,USA.mihir@cs.ucsd.edu.

[2] YuliangZheng. Digital signcryption or how to achieve cost (signature encryption) Cost (signature), Cost (encryption). In CRYPTO '97: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology, pages 165-179, London, UK, 1997. Springer-Verlag.

[3] F. Boo, R.H. Deng, "A signcryption scheme with signature directly verifiable by public key", Proceedings of PKC'98, LNCS 1431, Springer-Verlag, 1998, pp. 55- 59.

[4] Laura Suva, "Signcryption scheme based on Schnorr Digital Signature", Department of Information Security, Faculty of Mathematics and Computer Science, University of Bucharest, Bucharest, Romania. Proceedings of the IT Security for next generation, European Cup, Prague, 17-19 February, 2012.

[5] TAHER ELGAMAL,"A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms"IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-31, NO. 4, JULY 1985

[6] Guilin Wang1, Robert H. Deng1, DongJin Kwak2, and SangJae Moon2, " Security Analysis of Two Signcryption Schemes" Institute for Infocomm Research (I2R), 21 Heng Mui Keng Terrace, Singapore 119613.

[7] S. Sharmila Deva Selvi, S. Sree Vivek and C. Pandu Rangan,"Identity Based Public Verifiable Signcryption Scheme", Theoretical Computer Science Lab, Department of Computer Science and Engineering, Indian Institute of Technology Madras, India.

[8] Mohsen Toorani&Ali A. Beheshti, "An Elliptic Curve-based Signcryption Scheme with Forward Secrecy".

[9] William Stallings. Cryptography and Network Security: Principles and Practices. Prentice Hall Inc., second edition, 1999.

[10] Gamage, C., J. Leiwo, Encrypted message authentication by firewalls. Proceedings of International Workshop on Practice of Theory in Public Key Cryptography, Berlin, 69-81, 1999.

[11] Jung. H. Y, K.S Chang, D.H Lee and J.I. Lim, Signcryption scheme with forward secrecy. Proceeding of Information Security ApplicationWISA, Korea, 403-475, 2001.

[12] X. Yang Y. Han and Y. Hu. Signcryption based on elliptic curve and its multi-party schemes. Proceedings of the 3rd ACM International Conference on Information Security (InfoSecu 04), pages 216-217, 2004.

[13] Henri Cohen and Gerhard Frey, editors. Handbook of elliptic and hyperelliptic curve cryptography. CRC Press, 2005.

[14] Mohsen Toorani and Ali Asghar Beheshti Shirazi. Cryptanalysis of an efficient signcryption scheme with forward secrecy based on elliptic curve. Computer and Electrical Engineering, International Conference on, 428-432, 2008.

[15] Mohsen Toorani and Ali Asghar Beheshti Shirazi. An elliptic curve based signcryption scheme with forward secrecy. Journal of Applied Sciences, 9 (6): 1025 -1035, 2009.

[16] C.P. Schnorr, Efficient identification and signatures for smart cards, in G. Brassard, Ed. Advances in Cryptology-Crypto '89, 239-252, Springer-Verlag, 1990. Lecture Notes in Computer Science, nr 435.