International Journal of Cryptography and Security
ISSN: 2249-7013 & E-ISSN: 2249-7021, Volume 3, Issue 1, 2013, pp.-41-46.
Available online at http://www.bioinfopublication.org/jouarchive.php?opt=&jouid=BPJ0000222

AN EFFICIENT ID-BASED PUBLIC VERIFIABLE SIGNCRYPTION SCHEME

SWAPNA G. AND REDDY P.V.*
Department of Engineering Mathematics, A.U. College of Engineering, Andhra University, Visakhapatnam- 530 003, AP, India.
*Corresponding Author: Email- vasucrypto@yahoo.com

Received: June 22, 2012; Accepted: January 10, 2013

Abstract- Signcryption is a cryptographic scheme that combines the functionalities of signature and encryption in a single logical step. In a conventional signcryption scheme, the message is hidden and thus the validity of the signcryption can be verified only after the unsigncryption process. Thus, a third party will not be able to verify the validity of the signcryption. Signcryption schemes that allow anyone to verify the validity of signcryption without knowledge of the message are called public verifiable signcryption schemes. Third party verifiable signcryption schemes allow the receiver of signcryption, to convince a third party that the signcryption is valid, by providing some additional information (other than the receiver's private key) along with the signcryption. In this paper we propose an efficient ID-based signcryption scheme that offers public verifiability and third party verification, based on bilinear pairings over elliptic curves. We prove that our scheme satisfies the security notions such as confidentiality and unforgeability with the assumptions that the CBDHP, CDHP respectively is intractable in the random oracle model.

Keywords- Identity-based cryptography, bilinear pairings, signcryption, unforgeability, public verifiability

Citation: Swapna G. and Reddy P.V. (2013) An Efficient Id-Based Public Verifiable Signcryption Scheme. International Journal of Cryptography and Security, ISSN: 2249-7013 & E-ISSN: 2249-7021, Volume 3, Issue 1, pp.-41-46.

Introduction

Confidentiality, integrity, authentication and non- repudiation are the important requirements for many cryptographic applications. Confidentiality is keeping information secret from all other than those who are authorized to see it. Integrity is ensuring that the information has not been altered by unauthorized entities. Authentication is the assurance that the communicating party is the one that it claims to be. Non-repudiation is preventing the denial of the previous commitments or actions. Encryption can achieve the confidentiality and digital signature can achieve the integrity, authentication, and non- repudiation. If we need to achieve simultaneously confidentiality, integrity, authentication and non- repudiation, a traditional approach is first to sign a message and then to encrypt it, called sign-then-encrypt or signature-then-encryption approach. In 1997,Zheng [1] proposed a new cryptographic primitive called signcryption that fulfills both the functions of digital signatures and public key encryption simultaneously, at a cost of significantly lower than that required by the traditional signature-then-encryption approach. Signcryption has to found many applications, such as electronic transaction protocol, mobile agent protocol, key management, and routing protocol. The original scheme in [1] is based on the discrete logarithm problem but not security proof is given. Zheng's original schemes were only proven secure by Beak et al. [2] who described a formal security model in a multi-user setting. In the above mentioned traditional signcryption schemes, the public key of a user are essentially a random bit string picked from a given set. So, the signcryption does not provide the authentication of the user by itself. This problem can be solved via a certificate, which provides an unforgeable and trusted link between the public key and the identity of the user by the signature of a certificate authority (CA), and there is a hierarchical framework that is called public key infrastructure (PKI) to issue and manage certificates. However, the certificates management, including revocation, storage, distribution, and the computational cost of certificates verification are the main difficulties against traditional PKI.

To simplify the key management procedures of traditional PKI, Shamir [3] proposed the concept of identity - based cryptography (IBC) in 1984. The idea of IBC is to get rid of certificates by allowing a user's public key to be any binary string that uniquely identifies the user. Examples of such strings include e-mail addresses and IP addresses. Several practical identity-based signature (IBS) schemes [13] have been proposed since 1984, but a satisfying identity-based encryption (IBE) scheme only appeared in 2001 [4]. It was devised by Boneh and Franklin and cleverly uses bilinear maps (the Weil or Tate paring) over super singular elliptic curves.

The first identity based signcryption scheme proposed by Malone Lee [5] in 2002. Since then, many identity based signcryption schemes have been proposed in literature [6-10]. Their main objective is to reduce the computational complexity and to design the more efficient identity based signcryption scheme. In conventional signcryption, the sender signs the message which is hide it the receiver's public key. Thus, only the receiver can decrypt the message using his /her private key and can verify the authenticity of the cipher text.

Normally, in a signcryption scheme, the message is hidden and thus the validity of the signcryption can be verified only after the unsigncryption process. Thus, a third party (who is the unaware of the receiver's private key) will not be able to verify whether a signcryption is valid or not. Public verifiable signcryption scheme is well motivated in the following scenarios.

One of the main applications of signcryption scheme is secure e-mail systems. Public verifiable signcryption schemes are applicable in filtering out the spam's in secure e-mail systems. The spam filter should be able to verify the authenticity of the signcrypted e-mail without knowing the message (i.e., check whether the signcryption is generated from the claimed sender or not). Here, if the signcryption does not satisfy the public verifiability, it can be considered a spam and can be filtered out. 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 information computable with his private key required for the signature verification.

In 2004, Chow et al. [7] proposed the first ID-based public verifiable signcryption scheme. In 2010, Selvi et al. [11] showed attacks on confidentiality and unforgeability of the chow et al. [7] scheme, and proposed a new ID-based signcryption scheme with public verifiability and third party verification. In 2011, Prashant Kushwah et al. [12] proposed another identity based public verifiable signcryption scheme with third party verification and forward security.

In this paper we present an efficient ID-based public verifiable signcryption (ID-PVSC) scheme with third party verification, using bilinear pairings over elliptic curves. The proposed scheme satisfies the security notions such as confidentiality and unforgeability with the assumptions that the CBDH and CDH problems are intractable.

The rest of the paper is organized as follows: Section 2 briefly explains the bilinear pairings and some computational problems on which our scheme is based. The syntax and security requirements of our ID-PVSC scheme are given in Section 3. We present our ID-PVSC scheme in section 4. The correctness, security and efficiency analysis of the proposed scheme are given in Section 5. Section 6 concludes this paper.

Preliminaries

In this section, we briefly review bilinear pairings and some computational problems.

Bilinear Pairings

Bilinear pairing is an important primitive and has been widely adapted in many positive applications in cryptography. Let G_1 be an additive cyclic group with a prime order q and G_2 be a multiplicative cyclic group with the same order q. G_1 is a subgroup of the group of points on an elliptic curve and P is the generator of $G_1 \cdot G_2$ is a subgroup of the multiplicative group over a finite field. A bilinear pairing is a map $\hat{e}: G_1 \times G_1 \to G_2$ which satisfies the following properties.

1. Bilinear: $\hat{e}(aP, bP) = \hat{e}(P,P)^{ab}$ for all $P \in G_1$ and $a,b \in Z_q^*$.
2. Non-degenerate: There exists $P,Q \in G_1$ such that $\hat{e}(P,Q) \neq 1$.
3. Computability: There exists an efficient algorithm to compute $\hat{e}(P,Q)$ for all $P,Q \in G_1$.

We call such a bilinear map ê as an admissible bilinear pairing, and the Weil pairing in elliptic curve can give a good implementation of the admissible pairing [4].

Computational Problems

Now, we give some computational problems which will form the basis of security for our ID-PVSC scheme.

Definition 1 (Computational Diffie-Hellman Problem CDHP): The CDHP in G_1 is such that given (P, aP, bP) with uniformly random choices of $a,b \in Z_q^*$, to compute ab. The CDH assumption states that there is no polynomial time algorithm with a non-negligible advantage in solving the CDHP.

Definition 2 (Computational Bilinear Diffie-Hellman Problem CBDHP): The CBDHP is such that given (P,aP,bP,cP) with uniformly random choices of $a,b,c \in Z_q^*$, to compute $\hat{e}(P,P)^{abc}$.

Syntax and Security Model for ID-PVSC Scheme

In this section, we give the syntax for identity based signcryption scheme (ID-PVSC scheme) which supports both public verifiability and third party verification. We also give the security model for our ID-PVSC scheme.

Syntax of ID-PVSC Scheme

Our identity based signcryption scheme consists of the following algorithms.

Setup (1^k): Given the security parameter k, the Private Key Generator (PKG) generates the master private key M_{sk} and public parameters *Params*. *Params* are made public while M_{sk} is kept secret by the PKG.

Extract (ID_i): Given an identity ID_i as input, the PKG executes this algorithm to generate the private key S_{ID_i} corresponding to ID_i and S_{ID_i} sends to the user ID_i through a secure channel.

Signcrypt $(M, ID_A, S_{ID_A}, ID_B)$: A sender with identity ID_A and private key S_{ID_A} in order to signcrypt a message M to a receiver whose identity is ID_B, runs this algorithm to generate the corresponding signcryption σ.

Unsigncrypt $(\sigma, ID_A, S_{ID_B}, ID_B)$: On receiving the signcryption σ from sender ID_A, the receiver with identity ID_B and the private key S_{ID_B} o f the receiver, the receiver executes this algorithm to obtain the message M, if σ is a valid signcryption of M from ID_A to ID_B or "Invalid", indicating that the signcryption is not valid.

Public-Verify (σ, ID_A, ID_B): This algorithm allows any third party to verify the authenticity of the signcryption *σ* without knowing the message used for the generation of the signcryption σ It takes the signcryption σ, the sender identity ID_A and the receiver identity ID_B as input and outputs "Valid", if σ is a valid signcryption or "Invalid", otherwise.

TP-Verify (φ, ID_A, ID_B): This algorithm allows the receiver ID_B to prove the authenticity of the signcryption σ to third party by providing additional information needed (other than the private key S_{ID_B}). This algorithm runs by the third party and takes *φ* (σ and additional information provided by ID_B), the sender identity ID_A and receiver identity ID_B as input, and outputs "Valid", if σ is a valid signcryption from ID_A to ID_B or "Invalid", otherwise.

Security Model for ID-PVSC Scheme

Definition 3: (Message confidentiality):

An ID-based public verifiable signcryption scheme is said to be indistinguishable against adaptive chosen cipher text attacks (IND-ID-PVSC-CCA2) if no polynomially bounded adversary has non-negligible advantage in the following game.

Setup: The challenger C runs setup algorithm with a security parameter *k* and sends the system parameters to the adversary A.

Phase1: The adversary A performs a polynomially bounded number of queries to C. The queries made by A may be adaptive, i.e. current query may depend on the answers to the previous queries. The various oracles and the queries made to these oracles are defined below:

- **Key Extraction Queries (Oracle $O_{Extract}(ID_i)$):** A chooses an identity ID_i; C computes private key $S_{ID_i} = O_{Extract}(ID_i)$ to response to A.
- **Signcryption Queries (Oracle $O_{Signcrypt}(M, ID_A, ID_B)$):** A produces a signer identity ID_A, the recipient identity ID_B and a message M. C computes $S_{ID_A} = O_{Extract}(ID_A)$ and generates the signcryption σ for the message M using S_{ID_A} by following the signcryption protocol and sends σ to A.
- **Unsigncryption Queries (Oracle $O_{Unsigncrypt}(\sigma, ID_A, ID_B)$):** A produces ID_A, ID_B and the signcryption σ as input to this algorithm and requests the unsigncryption of σ. The challenger C runs unsigncrypt algorithm on input (*σ*, ID_A, ID_B) and returns its output to A. The result of the unsigncryption will be "Invalid" if σ is not a valid signcryption. It returns the message M, if σ is a valid signcryption.
- **TP-Verify Queries (Oracle $O_{TP\text{-}Verify}(\sigma, ID_A, ID_B)$):** A submits the information *φ*, the sender identity ID_A and the receiver identity ID_B. C generates the private key corresponding S_{ID_B} to ID_B, unsigncrypts σ using S_{ID_B} and returns the information required for TP-Verify corresponding to σ, if σ is a valid signcryption returns "Valid" if σ is a proper and correct signcryption. "Invalid" otherwise.

Selection and Challenge: At the end of the phase-1, A chooses two equal length plaintext M_0, M_1 and a sender identity ID_A and the recipient identity ID_B on which he wants to be challenged, and submits them to C. However A should not have queried the private key corresponding to ID_B in phase-1. C now chooses $\delta \in_R \{0,1\}$ and computes σ* $= O_{Signcrypt}(M_\delta, ID_A, ID_B)$ and sends σ* to A. It is to be noted that the private key S_{ID_A} corresponding to the sender ID_A can be queried by A.

Phase-2: A is allowed to interact with C as in phase-1 with the following restrictions.

- A should not query the extract oracle for the private key corresponding to the receiver identity ID_B.
- A should not query the Unsigncrypt oracle with (σ*, ID_A, ID_B) as input, i.e., a query of the form $O_{Unsigncrypt}$ (σ*, ID_A, ID_B) is not allowed.
- **Output (Guess):** Finally A produces a bit δ^1 and wins the game if $\delta^1=\delta$ The advantage of A in the above game is defined by

Adv(A) = 2|Pr[$\delta^1=\delta$]–1| where Pr[$\delta^1=\delta$] denotes the probability that $\delta^1=\delta$.

The confidentiality game described above deals with insider security since the adversary is given access to the private key of the sender ID_A used for the challenge phase.

Definition 4 (Unforgeability): *An ID-Based sign- cryption scheme is said to be existentially unforgeable against adaptive chosen message attacks (EUF- ID-PVSC-CMA) if no polynomially bounded adversary has a non-negligible advantage in the following game.*

Setup: The challenger C runs the Setup algorithm with security parameter *k* and obtains public parameters *Params* and the master private key M_{sk}. C sends *Params* to the adversary A and keeps M_{sk} secret.

Training phase: The adversary A performs a polynomially bounded number of queries to C as in Phase-1 of confidentiality game.

Forgery: After a sufficient amount of training, A produces a signcryption (σ, ID_A, ID_B) to C. Here A should not have required the private key of ID_A during the training phase and σ is not the output of signcrypt oracle with (*M*, ID_A, ID_B) as input (M=$O_{Unsigncrypt}$ (σ, ID_A, ID_B)). A wins the game, if Unsigncrypt (σ, ID_A, ID_B) is valid. It is to be noted that the private key S_{ID_B} corresponding to the receiver ID_B can be queried by A.

The security model discussed above captures the notion of insider security since the adversary is provided access to the private key of the receiver with identity ID_B used for generating the signcryption σ during the forgery phase.

Proposed ID-based Public Verifiable Signcryption Scheme (ID-PVSC Scheme)

In this section, we proposed an ID-based signcryption scheme that offers public verifiability and third party verification. We call it as ID-PVSC scheme. The ID-PVSC scheme consists of the following algorithms.

Setup: Given a security parameter *k*, this algorithm chooses two groups G_1 and G_2 with the same order q. Let $\hat{e}: G_1 X G_2 \longrightarrow G_2$ be an admissible bilinear pairing. Let P be the generator of G_1. Randomly choose $s \in Z_q^*$ and compute public key $P_{pub}=sP.H_1,H_2,H_3,H_4$ are hash functions and they satisfy $H_1:\{0,1\}^* \rightarrow G_1, H_2: G_2 \rightarrow \{0,1\}^*, H_3:\{0,1\}^n \rightarrow \{0,1\}^*, H_4:\{0,1\}^n \times G_1^3 \rightarrow Z_q^*$.

(*E*, *D*) is a secure symmetric encryption scheme. Then the system parameters are $Params = \{k, n, G_1, G_2, P, \hat{e}, H_1, H_2, H_3, H_4, E, D\}$.

Key Gen / Key Extract: For every user with identity ID_i, the PKG uses his master key $s \in Z_q^*$ and user's public key $Q_{ID_i} = H_1(ID_i)$ t o compute the corresponding secret key of $S_{ID_i} = sQ_{ID_i}$ the user with identity ID_i.

Signcrypt $(M, ID_A, S_{ID_A}, ID_B)$: To produce a signcryption on the message M under the recipient with identity ID_B the signer with identity ID_A uses his secret key S_{ID_A} to respond as follows.

1. Pick $s \in Z_q^*$ and compute $U = xP \in G_1$
2. Compute $\hat{\alpha} = \hat{e}(P_{pub}, Q_{ID_B})^x \in G_2$
3. Compute $\alpha_2 = H_2(\hat{\alpha})$
4. Compute $h = H_3(M, \hat{\alpha}, U, Q_{ID_A}, Q_{ID_B})$
5. Compute\ $C = E_{\alpha_2}(M \| h)$

6. Compute $R = H_4(C, U, Q_{ID_A}, Q_{ID_B})$
7. Compute $V = S_{ID_A} + xRP_{pub}$
8. The resultant signcryption text (ciphertext) on message M is σ = (*C,U,V*)

Public Verify $(\sigma, Q_{ID_A}, Q_{ID_B})$

1. The verifier first computes $\bar{R} = H_4(C, U, Q_{ID_A}, Q_{ID_B})$.
2. If $\hat{e}(P, V) = \hat{e}(P_{pub}, Q_{ID_A} + \bar{R}U)$,

then return "Valid". Otherwise, return "Invalid".

Unsigncrypt $(\sigma, Q_{ID_A}, Q_{ID_B}, S_{ID_B})$

1. If public verifi- $(\sigma, Q_{ID_A}, Q_{ID_B}) \neq$ ability "*Valid*" output "*Invalid*". Otherwise,
2. Compute $\hat{\alpha}' = \hat{e}(U, S_{ID_B})$
3. Compute $\alpha'_2 = H_2(\hat{\alpha}')$
4. Compute $M' \,\|\, h' = D_{\alpha_2'}(C)$
5. Output $\varphi = (M', h', \hat{\alpha}', \sigma)$ i f $h = H_3(M, \hat{\alpha}, U, Q_{ID_A}, Q_{ID_B})$ else, return "Invalid".

TP-Verify (φ, σ, *ID*$_A$, *ID*$_B$):

1. If Public Verify $(\sigma, Q_{ID_A}, Q_{ID_B}) \neq$ "Valid" output "Invalid". Otherwise,
2. $\bar{\alpha}_2 = H_2(\hat{\alpha}')$
3. $\bar{M} \,\|\, \bar{h} = D_{\bar{\alpha}_2}(C)$
4. Accept σ and output "Valid" if $\bar{h} = H_3(\bar{M}, \hat{\alpha}', U, Q_{ID_A}, Q_{ID_B})$ and $\bar{h} = h'$. Otherwise, "Invalid".

Analysis of the Proposed ID-PVSC Scheme

In this section, we discuss the proof of correctness, security analysis and efficiency analysis of the proposed ID-PVSC scheme.

Proof of the Correctness

The following equations give the correctness of signature verification:

$$\begin{aligned}\hat{e}(P,\ V) &= \hat{e}(P,\ S_{ID_A} + x\bar{R}P_{pub}) \\ &= \hat{e}(sP,\ H_1(ID_A) + x\bar{R}P) \\ &= \hat{e}(P_{pub},\ Q_{ID_A} + \bar{R}U)\end{aligned}$$

Correctness of $\hat{\alpha}'$:

$$\begin{aligned}\hat{\alpha} = \hat{e}(P_{pub},\ Q_{ID_B})^x &= \hat{e}(xP_{pub},\ Q_{ID_B}) \\ &= \hat{e}(xP,\ sQ_{ID_B}) \\ &= \hat{e}(U,\ S_{ID_B}) = \hat{\alpha}'.\end{aligned}$$

Security Analysis

In this section, we will formally prove the confidentiality and unforgeability of the proposed ID-PVSC scheme in the random oracle model.

Unforgeability of ID-PVSC Scheme

Theorem 1: *The proposed ID-PVSC Scheme is unforgeable in the random oracle model with the assumption that the Computational Diffie-Hellman Problem is intractable.*

Proof**:** Given a random instance $(P, A = aP, B = bP) \in G_1$ of the computational Diffie-Hellmann problem (CDHP), where $a, b \in Z_q^*$. We are going to construct a probabilistic polynomial time turing machine Δ which use the attacker A as a subroutine in order to compute CDH solution *abP* in G_1. In the whole game, A will consult Δ for answers to the random oracles H_1, H_2, H_3, H_4 and Δ needs to maintain hash lists L_1, L_2, L_3 and L_4 that are initially empty and are used to keep track of answers to queries asked by A to oracle H_1, H_2, H_3 and H_4. We assume that hash functions H_1, H_2, H_3 and H_4 were queried before signcryption.

- **Setup:** algorithm Δ sets $P_{pub} = A = aP$ as public key of PKG and sends the system parameters to the attacker attacker A.
- **Training Phase:** during the signing phase, the adversary A is allowed to access the various oracles provided by Δ. A can get sufficient training before generating the forgery. The various oracles provided by Δ to A during training are as follows.

***H*$_1$-queries** $(O_{H_1}(ID_i))$: To respond H_1– queries, Δ maintains a hash list L_1 which consists of (ID, Q_{ID}, d, T). When A queries the oracle H_1 at point *ID*, Δ responses as follows:

1. If the query ID already exists in the list L_1, then Δ responses with $H_1(ID) = Q_{ID}$.
2. Otherwise, Δ picks a random number $T \in \{0,1\}$. If *T*=0 then Δ computes $Q_{ID} = dbP$ for a random $d \in Z_q^*$. If *T*=1 then Δ computes $S_{ID} = dP$ for a random $d \in Z_q^*$. Δ adds the tuple (ID, Q_{ID}, d, T) to the list L_1 and returns to A with $H_1(ID) = Q_{ID}$.

***H*$_2$-queries** $(O_{H_2}(\hat{\alpha}))$: To respond H_2–queries**,** Δ maintains a hash list L_2 which consists of $(\alpha_2, \hat{\alpha})$. When A makes a query $\hat{\alpha}$, with input Δ responses as follows.

1. If the query *ID* already exists in the list L_2 then Δ responses with $\alpha_2 = H_2(\hat{\alpha})$.
2. Otherwise, Δ picks a random number $\alpha_2 \in Z_q^*$ to add the tuple $(\alpha_2, \hat{\alpha})$ to the list L_2 and responds to A with $\alpha_2 = H_2(\hat{\alpha})$.

***H*$_3$-queries** $(O_{H_3}(M, U, \hat{\alpha}, Q_{ID_A}, Q_{ID_B}))$: To respond H_3–queries, Δ maintains a hash list L_3 which consists of $(M, U, \hat{\alpha}, Q_{ID_A}, Q_{ID_B})$. When A queries the oracle H_3 at the point Δ $(M, U, \hat{\alpha}, Q_{ID_A}, Q_{ID_B})$, responses as follows.

1. If the query $(M, U, \hat{\alpha}, Q_{ID_A}, Q_{ID_B})$ already exists in L_3 then Δ returns *r* from L_3.
2. Otherwise, Δ picks a new random number $r \in Z_q^*$ and add the tuple $(M, U, \hat{\alpha}, Q_{ID_A}, Q_{ID_B}, r)$ to the list L_3 and responds to A with $H_3(M, U, \hat{\alpha}, Q_{ID_A}, Q_{ID_B}) = r$.

***H*$_4$-queries** $(O_{H_4}(C, U, Q_{ID_A}, Q_{ID_B}))$: To respond H_4– queries, Δ maintains a hash list L_4 which consists of $(C, U, Q_{ID_A}, Q_{ID_B})$, Δ responds as follows.

1. If $(C, U, Q_{ID_A}, Q_{ID_B}, R)$ is available in the list L_4 then Δ retrieves R from the list L_4.
2. Otherwise Δ picks a new random number $\hat{r} \in Z_q^*$ and sets $R = \hat{r}$ to add the tuple $(C, U, Q_{ID_A}, Q_{ID_B}, \hat{r}, R)$ to the list L_4 and responds to A with $H_4(C, U, Q_{ID_A}, Q_{ID_B}, \hat{r}) = R$.

Key Gen / Key Extract queries (O*$_{Extract}$*(*ID*)): When A submits an identity *ID* to the extract oracle, Δ recovers the corresponding (*ID,T,d*) entry from the list L_1.

1. If *T*=0, then Δ outputs 'failure' and halts, because it is unable to answer the query legitimately.

2. Otherwise, if T=1 it means that $H_1(ID)$ was previously defined as $dp \in G_1$, Δ computes $S_{ID} = dP_{pub} = dA$ and returns to A.

Signcrypt Oracle ($O_{Signcrypt}$(*M, ID_A, ID_B*))

When A asks for Signcrypt query on a message M under the signer's identity ID_A and the receivers identity ID_B Δ responses as follows: Δ generates the signcryption σ by doing the following computations.

1. Randomly choose $\hat{r}, x \in Z_q^*$ and sets $U = xP - \hat{r}^{-1}Q_{ID_A}$
2. Sets $\alpha_2 = H_2(\hat{\alpha} = \hat{e}(U, S_{ID_B}))$ and $r = H_3(M, U, \hat{\alpha}, Q_{ID_A}, Q_{ID_B})$.
3. Computes $C = E_{\alpha_2}(M \Box r)$.
4. Sets $R = O_{H_4}(C, U, Q_{ID_A}, Q_{ID_B}) = \hat{r}$ and
5. Compute $V = x\hat{r}P_{pub}$ and stores (C, U, Q_A, Q_B, R) in the list L_4. Here it should be noted that if a similar entry exists in L_4, repeat the procedure by choosing different $\hat{r}$.
6. Finally send the ciphertext σ=(*U,V,C*) to A.

Correctness of V can be shown as follows:

$$\hat{e}(P_{pub}, Q_{ID_A} + RU)$$
$$= \hat{e}(P_{pub}, Q_{ID_A} + \hat{r}(xP - \hat{r}^{-1}Q_{ID_A}))$$
$$= \hat{e}(P_{pub}, Q_{ID_A} + x\hat{r}P - Q_{ID_A}))$$
$$= \hat{e}(P, x\hat{r}P_{pub}))$$
$$= \hat{e}(P, V).$$

Unsigncrypt Oracle ($O_{Unsigncrypt}$(*σ,ID_A,ID_B*)) :

When A makes an unsigncrypt query with a sender's identity ID_A, a recipient's identity ID_B, and a ciphertext (*U,V,C*), D follows the steps below.

1. First, obtain the secret S_{ID_B} key of the recipient by key extraction algorithm.
2. Then, check whether the signcryption or ciphertext (*U,V,C*) is valid by the recipient's secret key and returns the corresponding output $\varphi = (M', h', \hat{\alpha}', \sigma)$.

TP-Verify Oracle ($O_{TP\text{-}Verify}$(*σ,ID_A,ID_B*)):

When A makes query with σ as input Δ performs the following:

Δ does the computations as given in unsigncrypt oracle and returns $\varphi = (\sigma, M', \hat{\alpha}', Q_{ID_A}, Q_{ID_B})$, if σ is valid, else, return "Invalid".

Output: Finally, A outputs a forgery σ*=(*U**,*V**,*C**) under the signer's identity ID_A^* and the recipient's identity ID_B^*. Then Δ checks ID_A^* in the list L_1. If in $T_A^* \neq 0$ the list L_1. Then Δ outputs failure and stops. Otherwise, the forgery is successful. By Forking lemma, after replaying A with the same random tape, Δ can obtain another signcryption text $\sigma_1^* = (C_1^*, U_1^*, V_1^*)$. For the two signcryption texts σ* and σ_1^*, they satisfy the following relations: $V^* = S_{ID_A^*} + x^*R^*P_{pub}$ and $V_1^* = S_{ID_A^*} + x^*R_1^*P_{pub}$. Then we have $R_1^*V^* - R^*V_1^* = (R_1^* - R^*)S_{ID_A^*} = (R_1^* - R^*)d^*abP$. Thus we can solve the CDH problem as $abP = \frac{R_1^*V^* - R^*V_1^*}{(R_1^* - R^*)d^*}$.

Confidentiality of ID-PVSC Scheme

Theorem 2: *The proposed ID-PVSC Scheme satisfies the confidentiality property in the random oracle model with the assumption that the Computational Bilinear Diffie-Hellman Problem is intractable.*

Proof: For proving the confidentiality of the ID-PVSC scheme, A is allowed to interact with Δ, as given in section 3. Assume that the challenger Δ is provided with the CBDHP instance $P, \bar{a}P, \bar{b}P, \bar{c}P$ from G_1 and is supposed to generate the solution $\hat{e}(P, P)^{\bar{a}\bar{b}\bar{c}} \in G_2$. Assume that there exists an algorithm A (adversary), capable of breaking ID-PVSC-CCA2 security of the scheme in polynomial time Δ can make use of A to find the solution for the CBDHP instance.

Setup: In order to provide the system parameters to A, Δ uses the CBDHP instance to cook up the system parameters as given below: Choose G_1, G_2 as the underlying group and P as the generator of G_1. Choose $P_{pub} = \bar{a}P$. Publishes $<G_1, G_2, q, P, P_{pub}>$, Δ also maintains lists L_1,L_2,L_3,L_4, and L_{Sign}, consistency in giving the responses to the queries made by A to various oracles.

Phase-I: During phase-I of training, the adversary A is allowed to access the various oracles provided by Δ. A can get sufficient training before taking up the challenge. The various oracles provided by Δ to A during Phase-I are similar to the oracles described in training phase of unforgeability proof.

Challenge Phase: At the end of the phase-1 interaction A picks two messages $<M_0,M_1>$of equal length, the sender identity ID_A and the receiver identity ID_B, and submits to Δ. On getting this, Δ chooses a random bit $\delta \in \{0,1\}$ and generates the signcryption on m_δ as follows.

- Chooses a random $\hat{r} \in Z_q^*$ sets and $R^* = \hat{r}$.
- Picks a random $C^* \in_R \{0,1\}^*$.
- Stores the tuple $(C^*, U^*, R^*, Q_{ID_A}, Q_{ID_B})$.
- Computes $V^* = \hat{R}\bar{C}P_{pub} + S_{ID_A}$. This is equivalent to $V^* = R^*\bar{C}P_{pub} + S_{ID_A}$.
- Sets $\sigma^* = (U^*, V^*, C^*)$.

Δ provides σ* as the challenge signcryption to A.

Phase-II: Now, A Interacts with Δ as in Phase-I, but with the following restrictions:

- A should not query the private key corresponding to ID_B to the extract oracle.
- A should not query the unsigncryption of σ* with ID_A as a sender and ID_B as receiver.
- A should not query for the third party verification of σ* with ID_A as a sender and ID_B as receiver.

Here, it should be noted that for getting the message M_δ from σ*, A should have queried H_2 or H_3 oracle. If A has H_2 or H_3 oracle, then it leaves an entry $(\hat{\alpha}^*, \alpha_2)$ in list L_2, where $\hat{\alpha}^* = \hat{e}(U, S_{ID_B}) = \hat{e}(\bar{c}P, \bar{a}\bar{b}P) = \hat{e}(P, P)^{\bar{a}\bar{b}\bar{c}}$. I f A has queried the H_3 oracle, then A should have computed $\hat{\alpha}^* = \hat{e}(P, P)^{\bar{a}\bar{b}\bar{c}}$. This leaves an entry $(M, U, \hat{\alpha}^*, Q_{ID_A}, Q_{ID_B}, r)$ in the list L_3. Therefore, on receiving A's response, Δ ignores the result and picks an $\hat{\alpha}$ from the list L_2 or L_3 and returns it as the solution to the CBDHP instance.

Efficiency Analysis of ID-PVSC Scheme

We compare the major computational costs and communication overhead (the length of the ciphertext) of our ID-PVSC scheme with those of Chow et al. scheme [7], Selvi et al. scheme [11], and Prashant Kushwah et al. scheme [12] in the [Table-1]. We consider only the costly operations which includes point scalar multiplications in G_1 (mul in G_1), exponentiation in G_2 (exp in G_2), and pairing operations (P).

Table 1- Computation and Communication overheads of the proposed ID-PVSC scheme

Scheme	Signcryption			Unsigncryption			Ciphertext Overhead
	Mul. in G_1	Exps. In G_1	P	Mul. in G_1	Exps. In G_1	P	
Chow, et al. [7]	2	0	2	1	0	4	$\|G_1\|+\|M\|+\|Z_q^*\|$
Selvi, et al. [11]	2	1	1	0	0	4	$2\|G_1\|+\|M\|$
Prashant, et al. [12]	3	1	0	0	0	3	$3\|G_1\|+\|M\|$
Our scheme	2	1	1	1	0	3	$2\|G_1\|+\|M\|$

In case of computational efficiency, our scheme needs 3 pairing operations as well as in scheme [12]. But the schemes in [7, 11] needs 4 pairing operations. Since the pairing computation is the most time consuming, the proposed scheme is more efficient than the schemes [7] and [11]. The size of the ciphertext in our scheme is $2|G_1|+|M|$, which is same as in the schemes [7, 11] and is less than the size of the ciphertext in the scheme [12]. Thus, our scheme has less computational overhead than Chow et al., Selvi et al. schemes and lower communication overhead than the Prashant Kushwah et al. scheme [12].

Conclusion

We have proposed a new ID-based signcryption scheme with public verifiability and third party verification. This scheme uses the bilinear pairings over elliptic curves. We have proved that our scheme satisfies the confidentiality and the unforgeability in the random oracle model with the assumption that CBDHP and CDHP computationally hard. Our scheme is efficient in terms of computational cost when compared with Chow et al., and Selvi et al., schemes and has lower communication overhead when compared with Prashant Kushwah scheme.

References

[1] Zheng Y. (1997) *Advances in Cryptology*, CRYPTO, 1294, 165-179.

[2] Baek J., Steinfeld R. and Zheng Y. (2002) *Public Key Cryptography*, 2274, 80-98.

[3] Shamir A. (1985) *Advances in Cryptology*, CRYPTO, 196, 47-53.

[4] Boneh D. and Franklin M. (2001) *Advances in Cryptology,* CRYPTO, 2139, 213-229.

[5] Malone-Lee J. (2002) *Cryptology e-Print Archive*, Report 2002/098.

[6] Libert B. and Quisquater J.J. (2003) *IEEE Information Theory Workshop*, 155-158.

[7] Chow S.S.M., Yiu S.M., Hui L.C.K and Chow K.P. (2003) 6*th International Conference Information Security and Cryptology*, 2971, 352-369.

[8] Boyen X. (2003) 23*rd Annual International Cryptology Conference*, CRYPTO, California, USA, 2729, 383-399.

[9] Chen L. and Malone-Lee (2005) *Public Key Cryptography*, 3386, 362-379.

[10] Barreto P.S.L.M., Libert B., McCullagh N. and Quisquater J.J. (2005) *Advances in Cryptology*, ASIACRYPT, 3788, 515-532.

[11] Selvi S.S.D., Vivek S.S. and Rangan C.P. (2010) *Prov. Sec.* 4*th International Conference*, 6402, 244-260.

[12] Prashant K. and Lal S. (2011) *International Journal Of Computer Science & Technology*, 2(3), 513-518.

[13] Gorantla M.C., Gangi Setti R. and Saxena A. (2005) *Cryptology e-print Archive*, Report 2005/094.