IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 12, December 2015.
www.ijiset.com
ISSN 2348 – 796 – 7968

# Heterogeneous Identity-based to certificate less online/offline signcryption

**Benjamin Klugah-Brown[1] Anthony Mackitz Dzisoo[2] Abigail Akosua Addobea[3]**

[1]School of life science and Technology
University of Electronic Science and Technology of China
Chengdu, 611731, China
bklugah@gmail.com

School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, 611731, China
[2]mackitz@gmail.com
[3]abigailaddobea@ymail.com

**Abstract**
This paper investigates current heterogeneous signcryption and proposes an efficient online/offline heterogeneous signcryption scheme which provides confidentiality and authentication to achieve security goals in secure communication. The proposed efficient signcryption scheme can simultaneously achieve confidentiality, integrity, authentication, and nonrepudiation in a logical single step. The scheme requires that a sender in an identity-based cryptosystem (IBC) to send a message to a receiver in a certificateless cryptosystem (CLC). The scheme is proved to have indistinguishability against adaptive chosen ciphertext attacks (IND-CCA2) under the bilinear Diffie-Hellman and existential unforgeability against adaptive chosen messages attacks (EUF-CMA) under the Diffie-Hellman problem in the random oracle model. The computational time and communication cost was determined through the implementation of the proposed scheme.

***Keywords:*** *Heterogeneous, online/offline Signcryption, Certificateless, signcryption, Identity-based cryptography, Certificateless setting, Identity-based* setting.

## 1. Introduction

### 1.1 Background

In 1997, Zheng [1] from Monash University in Australia discovered a new cryptography primitive called “Signcryption”. Signcryption is a new paradigm in public key cryptography that simultaneously fulfils 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 followed by encryption. These qualities are good for providing high security and low cost for many applications such wireless sensor networks.

One may wonder why a single primitive should carry such advantage. However, Zheng and others have demonstrated, through concrete examples, that signcryption schemes can provide clear benefits over the traditional sequential composition of encryption and signature schemes. The following shows the above reason.

(1) It is possible to create signcryption schemes which are more efficient, in terms of computational complexity and

communication overhead, than the sequential composition of signature and encryption schemes.

(2) Many signcryption schemes require only a single key pair for each user, while the traditional approach requires two: one for encrypting and one for signing.

(3) Some signcryption schemes allow expensive cryptographic operations to be parallelized.

(4) In its most basic form, the sequential composition of signature and encryption schemes yields only a weakly secure signcryption scheme. With some slight modifications, it is possible to achieve a much stronger level of security. This example suggests that it is valuable to work inside the framework of signcryption whenever both authenticity and confidentiality are required.

(5) The availability of a signcryption primitive can simplify the design of cryptographic protocols which require both authenticity and confidentiality.

### 1.2 Related works

This section will look at the various works that have been done over the years and recent innovations aimed

at improving signcryption and heterogeneous techniques.

1.2.1 Public key cryptography

To support the authenticity of public keys in the public key cryptography, there are three main infrastructures namely; Public key infrastructure (PKI), Identity-based cryptography (IBC) and certificateless public key cryptography (CL-PKC).

In PKI system, a certificate is used to bind a public key to a user's identity through a registration process. When this process is controlled and combined with digital signing technologies, the certificate can give a strong guarantee that only the expected user can access messages enclosed in the certificate. PKI, particularly in combination with smartcards, can provide robust user authentication and strong digital signatures. The main problem with PKI has to do with the need to manage certificates, including revocation, storage, and distribution. Also, verifying the validity of certificates before using them poses a huge problem.

In an IBC [3] system, identities (such as email addresses, IP address, etc.) are used as public keys. Any user can communicate with any other user by using the recipient's identity as the public key. Private keys are generated by a private key generator (PKG) from a server master secret. These basic properties allow for a secure messaging environment where no user requires certificates, and users need know nothing other than their identities. This design immediately eliminates both the problems of pre-enrollment, as well as that of certificate revocation checking. The PKG authenticates the recipient as required by policy. Once the recipient is successfully authenticated, the PKG generates the required private key and sends it to the receiver of the message which allows him to view his message. Notably, the problem with IBC is the key escrow problem.

The concept of CL-PKC was proposed by Al-Riyami and Paterson [4], which combines the functionality of a PKI with IBC. To encrypt a message, a sender requires both the receiver's identity and a public key value produced by the receiver. Similarly, to decrypt a ciphertext, a receiver requires the partial private key corresponding to his identity (which is given to them by a key generation center) and the secret value corresponding to the distributed public key. In certificateless cryptography there is a kind of trusted third party which is called key generation center (KGC) which is different from IBC's PKG by way of not having control or access to user's private keys. The KGC provides the user with a partial private key which it computes from the users' identity and a master key.

Signcryption: The concept of signcryption introduced by Zheng [1] is a new paradigm in public key cryptography that simultaneously fulfils 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 then encryption" approach. This primitive also answers the question 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. Two of the most popular signcryption are; Identity-based signcryption and certificateless signcryption.

1.2.2 Heterogeneous signcryption

This is a method that makes use of two primitive to either signcrypt or unsigncrypt a message. This method is very effective as it assumes the combined security of both primitives. In 2010, Sun and Li [7] proposed two heterogeneous signcryption schemes. The first scheme allows a sender in the PKI to send a message to a receiver in the IBC. The second scheme allows a sender in the IBC to send a message to a receiver in the PKI. But their schemes are only secure against outsider attacks and did not provide non-repudiation function. A stronger concept was then proposed by An et al [6] which covers insider security, this idea states that if a sender's secret key is exposed, an attacker is still not able to recover the message from the ciphertext and also if a receiver's secret key is exposed, an attacker is still not able to forge a ciphertext. Therefore, to achieve the insider security, Huang et al [8] proposed a heterogeneous signcryption scheme that allows a user in the IBC to send a message to a receiver in the PKI. Notwithstanding these facts, both schemes where found not appropriate for systems with low computational power because the computational cost is high for signcrypting a message. Heterogeneous signcryption is very important in the sense that, it give an overall advantage since the combined effect of any two schemes is realized. One of the notable work was done by Li and Xiong [11] in 2013. In their scheme they allowed the sender to be in IBC while the receiver belongs to PKI and is very suitable to deliver security solution for integrating wireless sensor networks into the Internet of things.

1.2.3 Online/offline methods

This method splits the sender's part of the communication into two phases and requires that most of computations are carried out when the associated message is still unavailable on one part known as offline phase, and continues computation when message is finally available on the other part called the online phase. This method is very suitable for heterogeneous techniques. The choice of this method is to achieve confidentiality, unforgeability as well as non-repudiation in a way that reduces cost.

(1) Online/offline signature: The notion of online/offline digital signature was introduced by Even, Goldreich and Micali. In their signature schemes, the

ISSN 2348 – 796 – 7968

first phase is performed offline prior to the arrival of a message to be signed and the second phase is performed online after knowing the message. The online phase is typically very fast. Online/offline signatures are particularly useful for low power devices such as smartcard applications.

(2) Online/offline signcryption: This notion was introduced by An et al [6] in 2002. As in the case of online/offline signature schemes, online/offline signcryption schemes should satisfy a basic property that online computation should be performed very sufficiently. All expensive operations such as exponentiation or pairing computation should be conducted offline in the first phase of the scheme. The second phase is performed online, once the message is presented. However, their scheme did not give any concrete construction of online/offline signcryption, but focused mainly on establishing formal security model for signcryption and analysis of some generic constructions. The first concrete online/offline signcryption scheme was given by Zhang et al [10], which introduces the need to reduce cost in terms of computation through the technique of splitting the signcryption into offline/online phases.

1.2.4 Organization

This paper is divided into five Sections. We begin with an introduction to the concepts of the research. The rest of the sections as follows; the preliminary work is introduced in section 2. The proposed scheme is described in Section 3. Security and performance analysis is described in Section 4. And Section 5. Gives a conclusion to the paper.

## 2. PRELIMINARIES

In this chapter, we shall look at a brief introduction to the basic definition and properties of bilinear pairing.

### 2.1 Theoretical basics

We introduce the concept of bilinear map and some definitions bellow.

Let $G_1 =< P >$ be a cyclic additive group generated by $P$, whose order is prime $q$ with identity ∞, and let $G_2$ be a cyclic multiplicative group of the same order $q$ with identity 1. A bilinear pairing on $(G_1, G_2)$ is a map $\hat{e}: G_1 \times G_1 \rightarrow G_2$ that satisfies the following conditions:

(1)(Bilinearity) $\forall R, S, T \in G_1, \hat{e}(R+S, T) = \hat{e}(R,T)\,\hat{e}(S,T)$ and $\hat{e}(R, S+T) = \hat{e}(R,S)\,\hat{e}(R,T)$.

(2) (Non-degeneracy) $\hat{e}(P,P) \neq 1$.

(3) (Computability) $\hat{e}$ can be efficiently computed.

The following properties of bilinear pairings can be easily verified. Property (5) is another way of defining non-degeneracy. For all $S, T \in G_1$.

(1) $\hat{e}(S, \infty) = 1$ and $\hat{e}(\infty, S) = 1$.

(2) $\hat{e}(S, -T) = \hat{e}(-S, T) = \hat{e}(S,T)^{-1}$.

(3) $\hat{e}(aS, bT) = \hat{e}(S,T)^{ab}\ \forall a, b \in \mathbb{Z}_q^*$.

(4) $\hat{e}(S,T) = \hat{e}(T,S)$.

(5) If $\hat{e}(S,R) = 1\ \forall R \in G_1$, then $S = \infty$.

One consequence of the bilinearity property is that the DLP in $G_1$ can be efficiently reduced to the DLP in $G_2$. For, if $(P, Q)$ is an instance of the DLP in $G_1$. Where $Q = xP$, then $\hat{e}(P,Q) = \hat{e}(P, xP) = (P,P)^x$. Thus $log_P Q = log_g h$, where $g = \hat{e}(P,P)$ and $h = \hat{e}(P,Q)$ are elements of $G_2$.

Let $\hat{e}$ be a bilinear pairing on $(G_1, G_2)$. The bilinear Diffie-Hellman problem (BDHP) is the following: Given $P, aP, bP, cP$, compute $\hat{e}(P,P)^{abc}$. Hardness of the BDHP implies the hardness of the DHP in both $G_1$ and $G_2$.

First, if the DHP in $G_1$ can be efficiently solved, then one could solve an instance of the BDHP by computing $abP$ and then $\hat{e}(abP, cP) = \hat{e}(P,P)^{abc}$. Also, if the DHP in $G_T$ can be efficiently solved, then the BDHP instance could be solved by computing $g = \hat{e}(P,P), g^{ab} = \hat{e}(aP, bP), g^c = \hat{e}(P, cP)$ and then $g^{abc}$. Nothing else is known about the intractability of the BDHP, and the problem is generally assumed to be just as hard as the DHP in $G_1$ and $G_2$.

We note that the decisional Diffie-Hellman problem (DDHP) in $G_1$ can be efficiently solved. The DDHP is to decide whether a given quadruple $(P, aP, bP, cP)$ of elements in $G_1$ is a valid Diffie-Hellman quadruple, i.e., whether $cP = abP$.

### 2.2 Computational complexity assumptions

The security of proposed scheme is based on some well-studied problems that are assumed to be hard to compute efficiently.

Computational Diffie-Hellman assumption (CDH): $G_1$ is a group of prime order $q, P \in G_1$ is the generator of $G_1, \forall\ a, b \in \mathbb{Z}_q^*$ given $P, aP$ and $bP$, computing $abP$ is hard.

Bilinear Diffie-Hellman assumption (BDH): $G_1$ and $G_2$ are two cyclic groups of prime order $q, P \in G_1^*$ is the generator of $G_1$, $\hat{e}$ is bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_1, \forall a, b \in \mathbb{Z}_q^*$, given $P, aP, bP$ and $cP$, computing $\hat{e}(P,P)^{abc}$ is hard.

Decision bilinear Diffie-Hellman assumption (DBDH): $G_1$ and $G_2$ are two cyclic groups of prime order $q, P \in G_1^*$ is the generator of $G_1$, $\hat{e}$ is a bilinear map $\hat{e}: G_1 \times G_1 \rightarrow G_1, \forall a, b \in \mathbb{Z}_q^*$. Given $P, aP, bP$ and $cP \in G_2$, it is hard to decide whether

$\hat{e}(P,P)^{abc} = w$. Where $w = G_2$

Gap bilinear Diffie-Hellman assumption (Gap-BDH): Given $(P, aP, bP)$ of the BDH problem with the help of a DBDH oracle that is able to decide whether a tuple $(P, a'P, b'P, c'P)$ is such that $c' \equiv a'b' \ (mod\ p)$ or not.

## 3. HETEROGENEOUS ONLINE/OFFLINE SIGNCRYPTION

### 3.1 Generic scheme

This paper considers an instance where the senders belong to the IBC and receivers belong to the CLC.

#### *3.1.1 Syntax*

HOOSC scheme: The scheme combines the polynomial time steps used in both certificateless and identity-based signcryption satisfying the conditions for the above mentioned cryptographic primitive.

(1) **Setup** $(1^k)$. Given a security parameter $1^k$, the PKG and KGC runs the setup algorithm to obtain secrete keys $msk_1$, $msk_2$ respectively and also returns a global system parameters $params$ including $mpk_1$, $mpk_2$ representing master public keys of PKG and KGC respectively.

(2) **Extract-Partial-Private-Key** $(ID, msk_2, params)$. This is an algorithm run by the KGC in which the user submits $msk_2, params$ and a verifiable identifier string $ID \in \{0,1\}^*$ representing user's identity, and returns $D_{ID}$ as the partial secrete key.

(3) **Generate-User-Keys** $(ID, params)$. An algorithm run by a user to produce user's secret value $\lambda_{ID}$ by taking user's identity $ID$ and system parameter $params$ as input and also returns a public key $PK$. The user obtained $\lambda_{ID}$ and $PK$ is used to construct a full private key.

(4) **Set-Private-Key** $(D_{ID}, \lambda_{ID}, params)$. This is a deterministic algorithm run by a user to return a full private key $SK_{ID}$ when it takes as an input $D_{ID}$ and $\lambda_{ID}$.

(5) **Extract-key** $(ID)$. Here the user submits its identity to the PKG who uses the master secret key $msk_1$ and user's $ID$ to generate the corresponding private key $SK_{ID}$ in a secure way. The signcryption and unsigncryption algorithms are as follows: We assume the sender of the message with an identity $ID_A$ and receiver with an identity $ID_B$.

(6) **Offline-SC**$(ID_A, SK_{IDA}, params)$. Given the sender identity $ID_A$ and the private key $SK_{IDA}$, this algorithm outputs an offline signcryption $\sigma$. This is executed by the sender with identity $ID_A$.**Online-SC**$(m, ID_A, ID_B, \sigma)$ This algorithm takes as input a message $m \in M$, the sender identity $ID_A$, the receiver identity $ID_B$ and the offline signcryption $\sigma$ as input and outputs an online signcryption $\delta$. This algorithm is executed by the sender with identity $ID_A$

(7). **USC**$(\delta, SK_{IDB}, ID_B, PK_{IDB}, ID_A, params)$. This is a deterministic unsigncryption algorithm. On input of a ciphertext $\delta$, receiver's full private key $SK_{IDB}$, identity $ID_B$ and public key $PK_{IDB}$, the sender's identity $ID_A$ and public key $PK_{IDA}$ and the global parameters $params$, the algorithm outputs a plaintext $m$ or a failure symbol $\perp$.

As stated earlier, a signcryption scheme should be able to provide confidentiality and authentication as these form the basis for any typical encryption and signature scheme respectively. The setup game between an adversary $A$ and an oracle $O$ shows ciphertext indistinguishability providing confidentiality as required for the game.

#### *3.1.2 Security notions*

To capture confidentiality there are two games in which the adversary will interact with the sender and receiver. The IND-CCA2 for both type I in which adversary $A_I$ is an attacker which is a usual user of the system who is not in possession of the KGC's master secret key. But it is able to adaptively replace users' public keys with (valid) public keys of its choice and type II an adversary $A_{II}$ who is also an honest-but-curious adversary KGC who knows the KGC's master key. But cannot replace user's public keys.

The detailed notion is captured in the following security games bellow as in [26] with a variation since the game models both PKG and KGC:

IND-CCA2-I: This depicts a game in which $A_I$ interacts with the "challenger":

Initial: Given a security parameter setup $(1^k)$, the challenger gets $(params, mpk_1, mpk_2)$ and gives $params$ and $msk_1$ to the adversary, while keeping master secret keys $msk_2$ to itself.

Phase 1: The adversary $A_I$ is allowed to adaptively perform a polynomially bounded number of queries.

(1) Extract partial private key: The adversary $A_I$ selects an $ID$ and sends it to the challenger. The challenger uses **Extract-Partial-Private-Key** $(params, msk_2, ID)$ algorithm to get $D_{ID}$ and sends it to the adversary.

(2) Extract private key: The adversary select an identity $ID$. With the challenger's computed $D_{ID}$, it uses the **Generate-User-Keys** $(ID, params)$ algorithm to get $(\lambda_{ID}, PK_{ID})$. Finally, it sends the result of $SK_{ID}$ computed from **Set-Private-Key** $(\lambda_{ID}, D_{ID})$ to the adversary. The adversary is not allowed to query any identity for which the corresponding public key has been replaced.

(3) Request public key: The adversary $A_I$ chooses an identity $ID$. The challenger gets $(\lambda_{ID}, PK_{ID}) =$

**Generate-User-Keys** $(params, ID)$ and sends $PK_{ID}$ to the adversary.

(4) Replace public key: The adversary may replace a public key $PK_{ID}$ with a value chosen.

(5) Unsigncryption queries: the adversary chooses $\delta$, a sender's identity $ID_A$ and a receiver's identity $ID_B$, the challenger finds $SK_{IDB}$ from its "query-answer" list, runs **Unsigncrypt** $(params, \delta, ID_A, SK_{IDB}, ID_B, PK_{IDB})$, and returns the result to the adversary . The result is either a plaintext message $m$ or $\perp$. Note that it is possible that the challenger is not aware of the receiver's secret value, if the associated public key has been replaced. In this case, we require the adversary to provide it. We also disallow queries where $ID_A = ID_B$.

Challenge: The adversary $A_I$ decides when Phase 1 ends. $A_I$ generates two equal length plaintexts $(m_0, m_1)$, a sender's identity $ID_A^*$ , and a receiver's identity $ID_B^*$ on which it wishes to be challenged. Note that $ID_B^*$ should not be queried to extract a private key in Phase 1. Furthermore, $ID_B^*$ cannot be equal to an identity for which both the public key has been replaced and the partial private key has been extracted. The challenger picks a random bit μ from {0,1}, computes

$\delta^*$ = Signcrypt $(params, m_\delta, SK_{IDA^*}, ID_{A^*}, ID_B, PK_{IDB^*})$, and returns $\delta^*$ to the adversary.

Phase 2: The adversary $A_I$ can ask a polynomially bounded number of queries adaptively again as in Phase 1. The same rule is applied here: $A_I$ cannot extract the private key for $ID_B$ . $A_I$ cannot extract the partial private key for $ID_B$ if the public key of this identity has been replaced before the challenge phase. In addition, $A_I$ cannot make an unsigncryption query on $\delta^*$ under $ID_A$ and $ID_B$ , unless the public key $PK_{ID_B^*}$ has been replaced after the challenge phase.

Guess: $A_I$ produces a bit $\mu_0$ and wins the game if $\mu_0 = \mu$.

The advantage of $A_I$ is defined to be;

$$Adv_{CLSC}^{IND-CCA2}(A_I) = |2\Pr[\mu' = \mu] - 1$$

where $\Pr[\mu_0 = \mu]$ denotes the probability that $\mu_0 = \mu$.

IND-CCA2-II: This is the game in which $A_{II}$ interacts with the challenger:

Initial: The challenger gets $(params, msk_1)$ Setup $(1^k)$ and gives both $params$ and $msk_1, msk_2$ to the adversary.

Phase 1: The adversary $A_{II}$ can perform a polynomially bounded number of queries in an adaptive manner. Note that we do not need extract partial private key since the adversary can compute partial private keys by itself.

(1) Extract private key: Same to the IND-CCA2-I game.
(2) Request public key: Same to the IND-CCA2-I game.
(3) Unsigncryption queries: Same to the IND-CCA2-I game.

Challenge: The adversary $A_{II}$ decides when Phase 1 ends. The adversary get two equal length plaintexts $(m_0, m_1)$, a sender's identity $ID_A$ , and a receiver's identity $ID_B$ on which it wishes to be challenged. $ID_B$ should not be queried to extract a private key in Phase 1. The challenger then selects a random bit $\mu$ from {0, 1} and gets

$\delta^*$ = **Signcrypt** $(params, m_\mu, SK_{ID_A^*}, ID_A, ID_B, PK_{ID_B^*})$, and returns $\delta^*$ to the adversary.

Phase 2: The adversary $A_{II}$ can ask a polynomially bounded number of queries adaptively again as in Phase 1. The adversary cannot extract the private key for $ID_B^*$. In addition, the adversary cannot make an unsigncryption query on $\delta^*$ under $ID_A^*$ and $ID_B^*$, unless the public key $PK_{ID_B^*}$ has been replaced after the challenge phase.

Guess: the adversary produces a bit $\mu_0$ and wins the game if $\mu_0 = \mu$. The advantage of $A_{II}$ is defined to be;

$$Adv_{CLSC}^{IND-CCA2-II}(A_{II}) = |2\Pr[\mu' = \mu] - 1|$$

where $\Pr[\mu_0 = \mu]$ denotes the probability that $\mu_0 = \mu$.

**Definition 1**. A CLC scheme is said to be IND-CCA2-I secure (resp. INDCCA2-II secure) if there is no probabilistic polynomial time (PPT) adversary $A_{II}$ (resp.$A_I$ ) which wins IND-CCA2-I (resp. IND-CCA2-II) with non-negligible advantage. A CLC scheme is said to be IND-CCA2 secure if it is both IND-CCA2-I secure and IND-CCA2-II secure. Notice that the adversary is allowed to extract the private key of $ID_A$ in the IND-CCA2-I and IND-CCA2-II games. This condition corresponds to the stringent requirement of insider security for confidentiality of signcryption. On the other hand, it ensures the forward security of the scheme, i.e. confidentiality is preserved in case the sender's private key becomes compromised.

For the strong existential unforgeability, "sUF-CMA" where adversary $F$ interact with their "challenger". Note that the challenger keeps a history of "query-answer" while interacting with the attackers. The game are described as follows;

sUF-CMA: Note that the adversary $F$ is required to have no knowledge about the environment it is querying when it interacts with the "challenger":

Initial: The challenger runs the setup algorithm $(params, msk_1)$; Setup $(1^k)$ and gives $msk_2, params$

ISSN 2348 – 796 – 7968

to the adversary. The challenger keeps master secret key $msk_1$ to itself.

Attack: The adversary performs a polynomially bounded queries as below;

(1) Extract private key: challenge algorithm run the **Extract-Key** algorithm $(params, msk_1, ID_A)$ and computes the corresponding private key $SK_{IDA}$ which it gives to $F$.

(2) Request public key: The adversary $F$ is allowed to make queries for any identity $ID$. The challenger gets $PK_{ID}$ by running $(\lambda_{ID}, PK_{ID}) =$ **Generate-User-Keys**. The challenger sends $PK_{ID}$ to $F$.

(3) Signcryption query: The challenge algorithm gets $SK_{IDA}$ and computes $\delta$ as the signcryption value when $F$ selects a message $m$, gets $ID_A$ and $ID_B$. The challenger sends $\delta$ to adversary $F$.

(4) Unsigncryption query: In this query algorithm $F$ selects a signcryption value $\delta$, senders identity $ID_A$ and receiver's identity $ID_B$ .The challenger computes unsigncryption and returns the results $m$ or $\perp$ to $F$

Forgery: $F$ produces a quaternion $(m^*, \delta^*, ID_A^*, ID_B^*)$. Note that $ID_A^*$ should not be queried to extract a private key. Note also that $ID_A^*$ cannot be equal to an identity for which both the public key has been replaced and the partial private key has been extracted. In addition, $\delta^*$ was not returned by the signcryption oracle on the input $(m^*, \delta^*, ID_A^*, ID_B^*)$ during the attack stage. $F$ wins the game if;

**Unsigncrypt**
$(params, \delta^*, SK_{ID_A^*}, ID_A, PK_{ID_A^*}, ID_B^*, PK_{ID_B^*})$ is not the $\perp$ symbol. The advantage of $F_I$ is defined as the probability that it wins.

**Definition 2**. A CLSC scheme is said to be sUF-CMA secure if there is no PPT adversary which wins sUF-CMA with non-negligible advantage. Note that the adversary is allowed to extract the private key of $ID_B$ in the above definition. Again, this condition corresponds to the stringent requirement of insider security for signcryption.

### 3.2 The proposed scheme

In this section, we propose an efficient heterogeneous online/offline signcryption scheme. The proposed scheme allows the sender in identity-based environment to sends a message to a receiver in certificateless environment. Here we will denote the sender who is in the sensor node as Alice (A), while the receiver Bob (B) in the Internet host.

#### *3.2.1 Description of the scheme*

We now present the online/offline identity-based-certificateless signcryption scheme bellow;

**Setup:** We choose four cryptographic hash functions:

$$H_1: \{0,1\}^* \rightarrow G_1$$
$$H_2: \{0,1\}^* \rightarrow \{0,1\}^n$$
$$H_3: \{0,1\}^* \rightarrow \mathbb{Z}_q^*$$
$$H_4: \{0,1\}^* \rightarrow \mathbb{Z}_q^*$$

Table 1: Symbols and meaning

| Symbols | Meaning |
|---|---|
| $ID_A$,$ID_B$ | Identities of sender and receiver respectively |
| $SK_{IDA}, SK_{IDA}$ | Private keys of sender and receiver rsp. |
| $msk_1, mpk_1$ | Master secrete and public keys (PKG) |
| $msk_2, mpk_2$ | Master secrete and public keys (KGC) |
| $s_1$ | Master Secret value from $\mathbb{Z}_q$(PKG) |
| $s_2$ | Master Secret value from $\mathbb{Z}_q$((KGC) |
| $\lambda$ | Secret value from a user (KGC) |
| $\alpha, \nu$ | Random values from $\mathbb{Z}_q$ |

Since the communication is between identity-based and certificateless environments, we prefer that the PKG and the KGC select different master secret values, such that the PKG selects $s_1 \in_R \mathbb{Z}_q^*$ as the master secret key $msk_1$ and KGC selects $s_2 \in_R \mathbb{Z}_q^*$, as the master secret key $msk_2$. The PKG sets $P_{pub1} = s_1 P$ and KGC set $mpk_2 = s_2 P$ respectively. The KGC inputs $(ID, msk_2)$ and using partial secret key extraction algorithm returns $D_{ID} = s_2 H(ID)$. The KGC choose a random value $s_2 \in \mathbb{Z}_q^*$. Let $P_{pub2} = s_2 P$ ($P$ is a random generator of $G_1$) be the known key of the system. The public parameter $params: < G_1, G_2, p, P_{pub}, H_1, H_2, H_3, H_4 >$.

**Extract Partial-Private-Key:** a user in the certificateless environment submits an identity $ID \in \{0,1\}^*$ to KGC which take as input $msk_2, params$. The KGC sets $D_{ID}$ and returns the partial private key as $D_{ID} = s_2 H(ID)$ in a secure way.

**Generate-User-Keys:** This an algorithm in which the user inputs it identity $ID$ and combining with the public parameter $params$ to return a secret value $\lambda_{ID}$ and a

ISSN 2348 – 796 – 7968

public key $PK_{ID}$ which can be used to construct the private keys of the user.

**Set-Full-Private-Key:** This is a deterministic algorithm run by the user in which a user submit $ID, \lambda_{ID}, params$, and gets $SK_{ID} = (\lambda, D_{ID})$.

**Extract-Key** $(ID)$. Here the user submits it identity to the PKG who generates the corresponding private key $SK_{ID} = s_1 H(ID)$ and given to user $ID$ in a secure way.

Assume that a sender's identity is $ID_A$ and the receiver's identity is $ID_B$.

**Off-Signcrypt:**

(1) $\alpha, v \in_R \mathbb{Z}_q^*$ compute $U = \alpha P$ , $T = \hat{e}(P_{pub2}, Q_{ID_B})^{\alpha}$

(2) Set $r = H_2(U, T, \alpha PK_{ID_B}, ID_B, PK_{ID_B})$

(3) Compute $\phi = SK_{ID_A} - vP$

(4) Output offline signcryption as $\sigma = (r, U, v, \alpha)$

**On-Signcrypt:**

(1) Compute $C = m \oplus r$

(2) Compute $h = H_3(U, C, ID_A)$

(3) Compute $\theta = \alpha h + v \bmod p$

(4) The ciphertext is $\delta = (U, C, \phi, \theta)$

**Unsigncryption:**

(1) Compute $h = H_3(U, C, ID_A)$

(2) If $\hat{e}(P_{pub}, Q_{ID_A})\hat{e}(P+U)^h \neq \hat{e}(P, \phi)\hat{e}(P, \theta P)$ return $\perp$ other wise move to (3)

(3) Compute $T = \hat{e}(U, D_{ID_B})$

(4) Compute $r = H_2(U, T, \lambda_B U, ID_B, PK_{ID_B})$

(5) Recover $m = C \oplus r$

*3.2.2 Correctness for decryption*

We give the correctness of decryption as follows;

$$\hat{e}(U, D_{IDB}) = \hat{e}(P_{pub2}, Q_{ID_B})^{\alpha}$$
$$= \hat{e}\left(P_{pub2}, Q_{ID_B}\right)^{\alpha}$$
$$= \hat{e}(s_2 P, Q_{ID_B})^{\alpha}$$
$$= \hat{e}(P, sQ_{ID_B})^{\alpha}$$
$$= \hat{e}\left(\alpha P, sQ_{ID_B}\right)$$
$$= \hat{e}\left(\alpha P, D_{ID_B}\right)$$
$$= \hat{e}\left(U, D_{ID_B}\right)$$

The receiver can recover the message by computing $m = C \oplus r$.

*3.2.3 Correctness for signature verification*

The correctness of the scheme is verified in the following steps bellow;

$$\hat{e}(P, \phi)\hat{e}(P, \theta P) = \hat{e}(P, \phi + \theta P)$$
$$= \hat{e}(P, SK_{IDA} - vP + \theta P)$$
$$= \hat{e}(P, SK_{IDA} - vP + (\alpha h + v)P)$$
$$= \hat{e}(P, SK_{IDA} - vP + \alpha hP + vP)$$
$$= \hat{e}(P, SK_{IDA} + \alpha hP)$$
$$= \hat{e}(P, SK_{IDA})\hat{e}(P, \alpha hP)$$
$$= \hat{e}(P, sQ_{IDA})\hat{e}(P, U)^h.$$
$$= \hat{e}(P_{pub}, Q_{IDA})\hat{e}(P, U)^h$$

### 3.3 Communication model

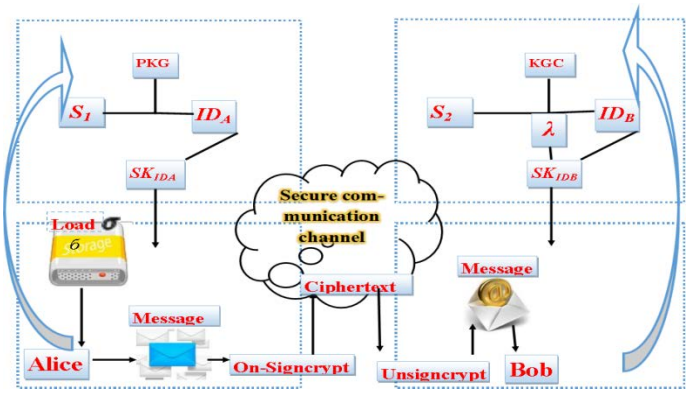


**Figure 1 Communication model for our scheme**

Figure 2. Procedure for secure communication based on the proposed scheme

Figure1. above shows the steps involved as Alice and Bob exchange messages through different media. As stated earlier, the scheme allows Alice to send a message an identity-based environment, while Bod belongs to a certificateless medium.

(1) The sensor node is loaded with precomputed results $\sigma = (r, U, v, \alpha)$ of the offline phase from a more powerful device. When the sender wants to send a message $m$ to a receiver, the sender runs $\delta =$ **On-Signcrypt**$(m, \sigma)$ algorithm and sends the ciphertext $\delta$ to the reciever. In this process, the sender only does light computations, such as exclusive OR, hash function and modular multiplication.

(2) When receiving the ciphertext $\delta$, the receiver runs $m =$ **Unsigncrypt** $(\delta, I\ D_A, SK_{ID_B}, ID_B)$

ISSN 2348 – 796 – 7968

algorithm to obtain the message $m$. The proposed scheme simultaneously achieves confidentiality, integrity, authentication and non-repudiation.

## 4. SECURITY AND PERFORMANCE ANALYSIS

### 4.1 Security analysis

The analysis of this scheme relies on the security proofs presented in [5] with a variation since the signcryption is done in identity-based environment.

**Theorem 1**. *The scheme above is IND-CCA secure, in the random oracle model, under the assumption that the gap bilinear Dffie-Hellman problem is intractable in the underlying bilinear group.* proof: this proof can be obtained by the following two lemmas

**Lemma 1.** *The GBDH assumption, states that, no PPT attacker A has non-negligible advantage in winning the IND-CCA game against the scheme proposed above, when all hash functions are modelled as random oracles there exists an algorithm B which uses A to solve the GBDH problem such that:.*

$$Adv_{IBC-CLC}^{IND-iCCA-I}(A) \leq q_T Adv_{\zeta}^{GBDH}(B, q_D^2 + 2qDq2)$$

where $q_T = q_1 + q_X + q_{SK} + 2_{qD} + 2$ . Here $q_T$ represent limit of queries carried by the adversary on following oracles; $H_1, H_2$, partial private key extraction, private key extraction and unsigncryption.

**Proof**. The challenge algorithm takes $(P, aP, bP, cP)$ as the GBDH challenge tuple with generator $P$. It sets $mpk_2 = cP$ and the system parameters $params$ including $(mpk_1, mpk_2)$ and makes it available to adversary $A$. The challenge algorithm selects $\ell = \in_R \{1 \dots q_T\}$ and responses to the various queries in $q_T$ as below;

$H_1$ **Queries:** let $ID_\ell$denote the $\ell^{th}$ non-repeated queried identity on $H_1$. The challenge algorithm selects $\alpha \in_R \mathbb{Z}_p$ if $v \neq \ell$ on the $v^{th}$ non-repeated query. It then sets $Q_{ID} = \alpha P$ and adds $(v, ID, \alpha)$ to an empty list $L_1$ while $Q_{ID}$ is returned. If $v = \ell$, the challenger produces $Q_{ID} = bP$ and adds $(v, ID, \perp)$ to $L_1$.

**Extract Partial Secret Key Queries:** The challenge algorithm obtains $(v, ID, \alpha)$ by calling $H_1$ on any new query $ID$. It then aborts the simulation if $v = \ell$, but returns $D_{ID} = \alpha cP$ if $v \neq \ell$.

**Extract Private Key Queries:** The challenge algorithm obtains $(v, ID, \alpha)$ by calling $H_1$ on any new query $ID$. It then aborts the simulation if $v = \ell$. The challenger is expected to possess an updated list $L_1$ containing a tuple $(ID, PK, \perp)$ upon an input of $ID$and$PK$. If $v \neq \ell$, the challenger searches $L_1$ for entry $(ID, PK, \lambda)$, it then proceeds to get new key pair and returns $(\lambda, \alpha cP)$ if the tuple entry does not exist.

$H_3$ **Queries:** The challenge algorithm selects a value $z \in_R \mathbb{Z}_p$ for every new query tuple$(U, C, ID)$. It updates the empty list $L_3$ with the values $z$ and $zP$ and finally returns $zP$.

$H_2$ **Queries:** The challenge algorithm performs the following for each new query $(U, T, \vartheta, ID, PK)$ where $\vartheta = \lambda U$;

(1) The challenger confirms the consistency of the DBDH algorithm on the queriedtuple $(aP, bP, cP, T)$ if it returns 1. $T$ is produced and the algorithm finally stops if the confirmation returns true.

(2) The challenger now searches for different hash value $r$ in $M_2(U, \vartheta, ID, PK, r)$ in such a way that if the tuple $(U, bP, cP, T)$ is given, the DBDH oracle returns 1. $r$ is then returned if such a tuple exist. Note that in this case $ID = ID_{\ell}$.

(3) The challenger returns $r$ and updates an empty list $L_2$ with the returned values and the input contained in the tuple.

**Unsigncryption Queries:** The challenge algorithm performs the following task on each query $(U, C, \theta, ID_A, ID_B)$;

(1) The challenger obtains $Q_{ID}$ and $PK$ by running $h$ and request public key oracle and returns $\perp$ if verification is does not exist.

(2) $(w, ID_B, \alpha)$ is obtained by calling $H_1$ on $ID_B$ when $T = \hat{e}(\alpha U, mpk_2)$ is computed and calls $H_2$ to complete the Unsigncryption. Note, this is only possible if $ID_B \neq ID_\ell$.

(3) The challenger then searches for different values of $T$ in $(U, T, \vartheta, ID_\ell, PK_B, r)$ contained in $L_2$, such that when there is a query on $(U, bP, cP, T)$, the DBDH oracle returns 1. This is done to prove the consistency in the answers of the challenger, since pairing cannot be computed if $ID_B = ID_\ell$ . The challenger proceeds to decrypt using the hash value $r$ when the correct pairing value is found.

(4) The challenger places the entry $(U, \vartheta, ID_\ell, PK_B, r)$ for a random $r$ on list $L_2$ at this stage and decrypting using the hash value $r$.

Challenge: At this stage the challenge algorithm places a query on $H_1$ by getting $ID_B^*$ when the adversary outputs two messages $m_0$ and $m_1$ (assuming they are of equal length). The adversary get the two identities $ID_A^*$ and $ID_B^*$ on which it hope to be challenged. The challenge algorithm aborts if $ID_B^* \neq ID_\ell$, otherwise, it searches the list $L_1$ containing the pair $(PK_{IDA}, ID_A)$ and sets $U^* = aP$. The challenger proceeds as follows; it selects $\delta \in_R \{0,1\}^n$, a hash $r^* \in_R \mathbb{Z}_p^*$ and sets $C^* = m_\delta \oplus r^*$. The challenge algorithm sends $\delta$. to the adversary as the challenged ciphertext.

ISSN 2348 – 796 – 7968

Guess: Algorithm $A_I$ outputs a guess on the ciphertext but may not be able to identify the right signcrypted message unless it runs a query on $H_2$ the tuple $(U^*, T^*, \vartheta^*, ID_\ell, PK^*)$. The challenger wins the advantage if the challenged tuple cannot be found in $L_2$, also $L_1$ has at most $q_T$ elements with probability $\frac{1}{q_T}$. The adversary has no advantage for this case and otherwise.

**lemma 2**. *Under the CDH assumption in $G_1$ no PPT attacker A has non- negligible advantage in winning the IND-CCA2 game against the scheme proposed above, when all hash functions are modelled as random oracles. More precisely, there exists an algorithm B which uses A to solve the CDH problem such that:*

$$Adv_{IBSC-CLSC}^{IND-CAA}(A) \leq q_T Adv_{\zeta}^{CDH}(B)$$

where $q_T = q_{PK} + q_{RPK} + q_{SK} + q_{SK_{ID_B}} + 2_{q_D} + 2$ . Here $q_{PK}$ and $q_{RPK}$ are the maximum number of queries that the adversary could place to request public key and replace public key oracles and $q_{SK}$ and $q_D$ are as before.

**Proof**. Let $q_T$ be the statement of the lemma. The challenge algorithm takes the challenge tuple $(aP, cP)$ with the generator $P$ . The challenger selects $s_1, s_2 \in_R \mathbb{Z}_p^*$ as the master secret keys $msk_1$ and $msk_2$ and sets $mpk_1 = s_1 P$ and $mpk_2 = s_2 P$ as the master public keys respectively. The challenger gets the master secret key-public key pair $(msk_1, mpk_1)$ and $(msk_2, mpk_2)$ and gives $(msk_1, msk_2)$ to the adversary. The challenger selects an index $\ell \in_R \{1 \dots q_T\}$ and responses to the various queries in $q_T$ as follows:

$H_1$ **Queries:** The challenger selects $\alpha \in_R \mathbb{Z}_p$ and sets $Q_{ID} = \alpha P$. An empty list $L_1$ is updated with $(ID, \alpha)$. The challenger returns $Q_{ID}$ .

**Request Public Key Queries:** Let $ID_\ell$ denote the $\ell^{th}$ non-repeated queried identity. The challenger selects $\lambda \in_R \mathbb{Z}_p$ and generates a new key pair $(\lambda, PK)$. If $v \neq \ell$ on the $v^{th}$ non-repeated query, $L_2$ is updated with the tuple $(v, ID, \lambda, PK)$. The challenger produces $cP$ and updates $L_1$ with $(\ell, ID, cP, \perp)$ so long as $v = \ell$.

**Extract Private Key Queries:** The challenge algorithm obtains the tuple $(v, ID, PK, \lambda)$ by running the request public key on every new query on $ID$. The challenger proceeds so long as $v \neq \ell$ to return $(\lambda, msk_1 \alpha P)$ when it obtains $(ID, \alpha)$. The challenge algorithm does this by calling $H_1$ on $ID$. The challenger aborts the simulation if $v = \ell$.

$H_3$ **Queries:** The challenge algorithm selects a value $z \in_R \mathbb{Z}_p$ for every new query tuple $(U, C, ID)$. It updates the empty list $L_3$ with the values $z$ and $zP$ and finally returns $zP$.

$H_2$ **Queries:** The challenge algorithm performs the following task on each new query $(U, T, ID_A)$:

(1) The challenger produces $\vartheta$ and stops if $\hat{e}(bP, cP) = \hat{e}(P, \phi)$ .

(2) The challenger returns $r$ if $ID_A = ID_\ell$ such that $\hat{e}(U, bP) = \hat{e}(P, \phi)$ . It does this by searching through $L_2$ for the entry tuple $(U, T, ID, r)$.

(3) The challenger returns $r$ and updates an empty list $L_2$ with the returned values and the input contained in the tuple.

**Unsigncryption Queries:** The challenge algorithm performs the following tasks on every new query tuple $(U, C, \phi, ID_A, ID_B)$:

(1) The challenger obtains $Q_{ID}$ and $PK$ by running $H_1$ and request public key oracle and returns $\perp$ if verification is does not exist.

(2) $(ID_B, \alpha_B)$ is obtained by calling $H_1$ when $T = \hat{e}(U, \alpha_B P_{pub2})$ is computed, It continues to compute $\alpha_B P$ if $ID_B \neq ID_\ell$. The challenger finally gets $L_2$ to complete the unsigncryption stage.

(3) The challenger searches for different values of $\alpha_B P$ in the tuple $(U, T, ID_\ell, r)$ by going through $L_2$, such that the pairing, $\hat{e}(U, aP) = \hat{e}(P, \phi)$. This is done to check the consistency in the answers provided by the challenge algorithm since $\alpha' P$ cannot be calculated If $ID_B = ID_\ell$. The challenger proceeds to decrypt using the hash value $r$ when the correct value of $\alpha_B P$ is obtained. The challenger places the entry tuple $(U, T, ID_B, r)$ for a random $r$ on the list $L_2$. Finally decrypt using the value of $r$.

Challenge: At this stage the challenge algorithm places a query on $H_1$ by getting $ID_B^*$ when the adversary outputs two messages $m_0$ and $m_1$ (assuming they are of equal length). The challenger obtains $(\omega, ID_B^*, PK^*, \lambda)$ by calling request public key for $ID_B^*$ and aborts if $\omega \neq \ell$, else it runs request public key to obtain the pair $(PK, ID_A^*)$. The challenger sets $U^* \equiv aP$. The challenger proceeds as follows; it selects $\delta \in_R \{0,1\}^k$ , a hash $r^* \in_R \mathbb{Z}_p^*$ and sets $C^* \equiv m_\delta \oplus r^*$.

Guess: Algorithm $A_I$ outputs a guess on the ciphertext and wins the advantage if the challenged tuple cannot be found in $L_2$, also $L_1$ has at most $q_T$ elements with probability $\frac{1}{qT}$. However, the adversary may not be able to identify the right signcrypted message unless it runs a query on $H_2$ containing tuple$(U^*, T^*, \vartheta^*, ID_\ell, PK^*)$. The adversary has no advantage for this case and otherwise.

**Theorem 2**. *The scheme above is sUF-CMA secure, in the random oracle model, under the GDH assumption in $G_1$, which states that, no PPT attacker A has non-negligible advantage in winning the sUF-CMA game against the scheme proposed above, when all hash*

ISSN 2348 – 796 – 7968

*functions are modelled as random oracles. More precisely, there exists an algorithm B which uses A to solve the GDH problem such that:*

$$Adv_{IBSC-CLSC}^{SUF-iCMA-I}(A) \leq q_T Adv_{\zeta}^{GDH}\left(B, q_D^2 + 2q_{Dq_2}\right) + (q_{SC}\,(q_{SC} + q_D + q_3 + 1)/2^k$$

where $q_T = q_1 + q_X + q_{SK} + 2q_D + q_{qSC} + 1$. Here $q_T$ represent limit of queries carried by the adversary on following oracles; $H_3$ and signcryption oracles.

**Proof**. The challenge algorithm takes $(bP, cP)$ as the GDH challenge tuple with generator$P$. It sets $mpk_2 = aP$ and the system parameters $params$ including $(mpk_1, mpk_2)$ and makes it available to adversary. The challenge algorithm selects $\ell \in_R \{1 \ldots q_T\}$ and responses to the various queries in $q_T$ as below:

$H_1$ **Queries:** let $ID_\ell$ denote the $\ell^{th}$ non-repeated queried identity on $H_1$ the challenge algorithm selects $\alpha \in_R \mathbb{Z}_p$ if $v \neq \ell$ on the $\ell^{th}$ non-repeated query. It then sets $Q_{ID} = \alpha P$ and adds $(v, ID, \alpha)$ to an empty list $L$ while $Q_{ID}$ is returned. If $v = \ell$, C produces $Q_{ID} = bP$ and adds $(v, ID, \perp)$ to $L_1$.

**Extract-Key:** For each new query $ID$, algorithm recovers $L_1$ and returns $SK_{ID_v}$. If $v = \ell$, the challenger algorithm aborts the simulation. Otherwise it calls $H_1$ on $ID$.

$H_3$ **Queries:** The challenge algorithm generates a value $z \in_R \mathbb{Z}_p$ for every new query tuple $(U, C, ID)$. It updates the empty list $L_3$ with the values $z$ and $zP$ and finally returns $zP$.

$H_2$ **Queries:** The challenge algorithm performs the following for each new query$(U, T, \vartheta, ID, PK)$:

(1) The challenger gets $\vartheta$ and stops if $\hat{e}(bP, cP) = \hat{e}(P, \phi)$. The challenger now searches for different $r$ in $L_2$ $(U, \vartheta, ID, PK, r)$ in such away that if the tuple $(U, bP, cP, T)$ is given, the DBDH oracle returns 1. $r$ is then returned if such a tuple exist when $ID = ID_\ell$.

(2) The challenger returns $r$ and updates an empty list $L_2$ with the returned values and the input contained in the tuple.

**Signcryption Queries:** The challenge algorithm performs the following task on each updated query $(m, ID_A, ID_B)$:

(1) The challenge algorithm is able to signcrypt the message $m$ by calling $H_1$ on $ID$ and running the **Extract-Key** or from the adversary while $ID_A = ID_\ell$.

(2) The challenge algorithm computes $T = \hat{e}(U, \alpha_B mpk_1)$ and runs $H_1$ on $ID_B$ to get $(\omega, ID_B, \alpha_B)$. The challenger checks if the following; $ID_A = ID_\ell$ and $ID_B \neq ID_\ell$, is true, the challenger selects two values $x, y \in_R \mathbb{Z}_p$.

(3) The challenger now computes $C = m \oplus r$ if the following are true:

(a) Challenger searches for $\vartheta$ in $L_2$ containing the entry $(U, T, \vartheta, ID_B, PK_B, r)$

(b) Checks if $\hat{e}(U, PK_{IDB}) = \hat{e}(P, \vartheta)$.

Note: $PK_{IDB}$ is obtained by calling the request public key oracle on $ID_B$. If the above produces false the challenger updates the $L_2$ with $(U, T, ID_B, PK_B, r)$ using the random $r$.

(4) The challenge algorithm checks to see if different value has already been given in place of the defined $H_3(U, C, ID, PK)$ value which it sets as $H = y^{-1}(xP - Q_{ID})$. If such a value exist, the challenger aborts the simulation.

The adversary uses the two identities $ID_A^*$ and $ID_B^*$ to output a tuple $(U^*, C^*, \phi, \theta)$. The challenger performs the following; it runs $H_1$ on $ID^*$ and proceeds execution if $ID_A^* = ID_\ell$, aborts if otherwise. If $ID_A^* \neq ID_\ell$, the challenger calls on request public key oracle on $ID_A^*$ to obtain $PK^*$. The challenger gets $z$ from $L_3$ through $H_3$ oracle on $(U^*, C^*, ID_\ell, PK_B^*)$. The adversary is said to have won the game if; the challenge algorithm only fails to gain the advantage over the adversary. The adversary runs a partial and full private key extract on $ID$ and the challenger replays the oracle simulation on $H_3$.

### 4.2 Performance Analysis

In this section we compare the efficiency in the proposed scheme with three other schemes [5,8,2]. Note the abbreviations used in Table 2: CCA2 (adaptive chosen ciphertext attack), $Mult$ (point multiplication in $G_1$), $Exp$ (exponentiation in $G_2$) CMA (chosen message attack) and $\hat{e}$ (pairing computations).

ISSN 2348 – 796 – 7968

Table 2: Performance Comparison

| scheme | Security | | Performance | | |
|---|---|---|---|---|---|
| | CCA2 | CMA | Mult | Exp | $\hat{e}$ |
| M. Barbosa and P. Farshi | Yes | Yes | 5 | 1 | 6 |
| Q. Huang *et al* [8] | Yes | Yes | 2 | 2 | 2 |
| J. Malone – Lee[2] | No | Yes | 3 | 1 | 6 |
| Proposed scheme | Yes | Yes | 3 | 2 | 4 |

From the results in Table 2 above, it can been seen that, considering computational cost of pairing, the proposed scheme is more efficient than the schemes stated above, as the proposed scheme makes used of only one pairing operation during signcryption . The proposed scheme also splits the signcryption into two phases: offline phase and online phase. Two point multiplication operations have been pre-computed offline. The online phase is very efficient and does not need any pairing. This shows that the proposed scheme can complete quickly the entire signcryption process when a message is available. Therefore, the proposed scheme is very suitable to provide security solution for low capacity devices such as sensor nodes.

4.2.1 Implementation

This section shows the implantation using pairing based cryptographic library with C programing on a machine whose system specifications are given in Table 3 bellow. All execution time are calculated in seconds per CPU time. Let $G_1 = 1024$bit,

$G_2 =$ 1024 bits, a Type A pairings constructed on the curve $y^2 = x^3 + x \bmod q$, $p = 160$ bits $q = 512$ bits and a user $ID = 160$ bits. Using the following parameter;

$P$ =[155602052632388626893585481086564195412275577579913298313020768027296674473546665085459528137525063522082770830164280683828402406793656550297976947188285799060459815861824198551848615547834935277281893438383910690556267033584025553494156738591035595799362425040317124305951462057260058921195972839182148341]

Table 3: Machine Specification

| Machine | CPU | OS | Crypto Library |
|---|---|---|---|
| Dell N50 | T4500 at 2.30Ghz x 2 | 64 bit Win7 | Pairing based Crypto Library |

Table 4: Key Generation

| Message | $G_1$ | $G_2$ | $P_{pub}$ | $p$ | $q$ |
|---|---|---|---|---|---|
| 160 bit | 160 bit | 1024 bit | 1024 bit | 512 bit | 160 bit |

Table 5: Execution Time (ms)

| Messag | Signcryption | Verificatio | Overall (includin |
|---|---|---|---|
| 160 bits | 0.072 | 0.064 | 0.21 |

4.1.1 Simulation results

The Figure 2 shows the results obtained from simulation of an 8 character message and parameters from PBC library as shown in Table 4. The time for signcryption, verification and the ovarall execution is shown in the output environment calculated in second per CPU time. Figure 3. Show the execution environment of system

ISSN 2348 – 796 – 7968

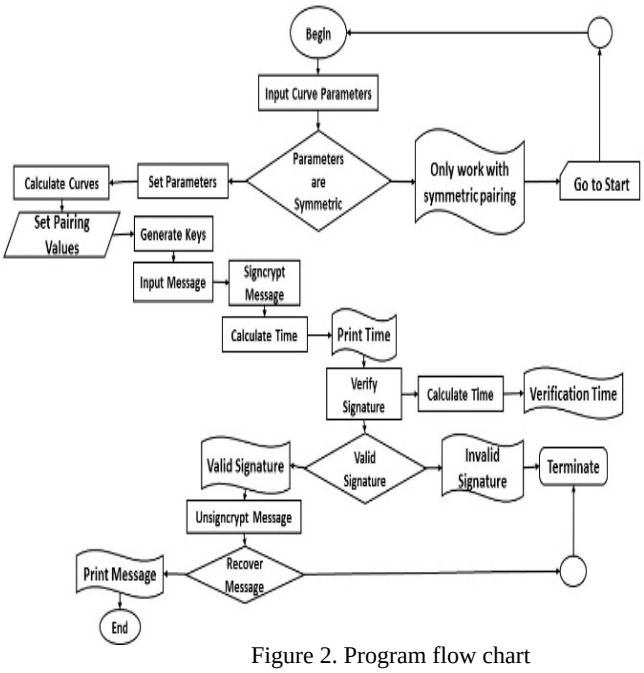


Figure 2. Program flow chart

Figure 2, shows the programing logic of the implementation from “Begin” to “End” shown if the steps below;

(1) The program begins with an input of curve parameter, in this case is Type A curve of PBC library.
(2) The state of the parameters are assed to check if it falls within the concept of symmetricity, if “yes” the user continues to set variable, else the correct symmetric parameter re-inputted.
(3) With the parameters and variables set, the program proceeds to calculate the curve based of the type of curve, in this case it is $y^2 = x^3 + x \bmod q$ followed by generations of key for signcryption.
(4) Message is then inputted for signcryption, time is then calculated and printed out.
(5) Signature verification is done and time is also calculated and printed out.
(6) The verification is assessed, if the signature is valid
(7) The programme proceeds, else the programme terminates.
(8) Unsigncryption is done once the signature is valid
(9) Recovery algorithm is called, if message can’t be recovered the programme terminate, else the recovered message is printed.
(10) The programme ends

Figure 3. Shows the programed output in a console environment with C programing, it indicates key generation with their value in reference to the type of curve used, the message to be signcrypted and time used. The signcrypted value is shown and verified with time calculated. Finally the total time is computed and printed including key generation and computational time over the non-heterogeneous schemes. In this scheme the sender need not worry about computational cost since in the offline mode there in no message while the receiver can do huge computations during the recovery of message. However, the receiver’s computation is also reduced since it need not worry about certification which is usually accompanied by huge computation and delays as inherent in PKI

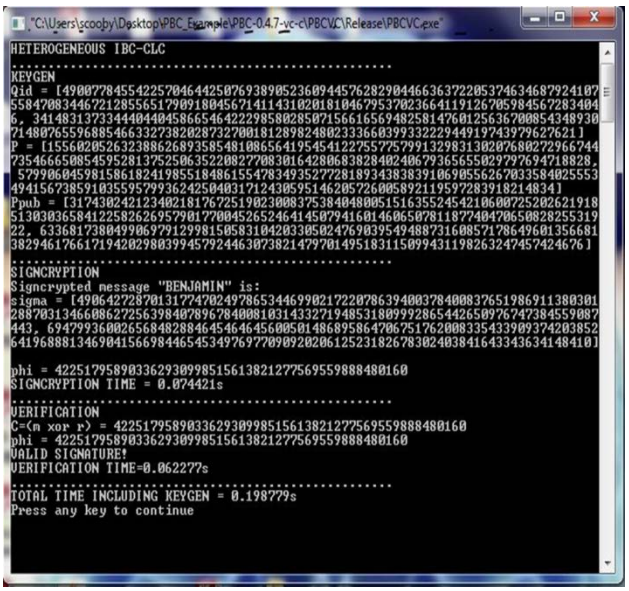


Figure 3. Simulated results degenerated from signcrypted message

ISSN 2348 – 796 – 7968

# 5. CONCLUSION

## 5.1 Concluding remarks

Heterogeneous online/offline signcryption has recently gain much attention as it provides a much higher potentials of achieving all the security characteristic within a single logical step. The proposed heterogeneous online/offline scheme designed in this paper has shown to provide a higher security. The combined advantage of online/offline and lower pairing operations proposed in this paper gives solution to the security and suitable for integrating a WSNs into the IoT.

# 6. REFERENCES

[1] Y. Zheng, Signcryption and Its Applications in Efficient Public Key Solutions, Information Security Workshop (ISW'97), Springer-Verlag, pp.291-312, 1998.

[2] J. Malone-Lee, Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002.

[3] A.Shamir, Identity-Based Cryptosystems and Signature Schemes. Advances in Cryptology, LNCS 196, Springer-Verlag, pp.47-53, 1985.

[4] S.S. Al-Riyami and K.G. Paterson, Certificateless Public-Key Cryptography. Advances in Cryptology-ASIACRYPT, LNCS 2894: pp.452-473, 2003.

[5] M. Barbosa and P. Farshim, Certificateless Signcryption. Proceedings of the 2008 ACM symposium on Information, computer and communications security, pp. 369 372, 2008.

[6] An J H, Dodis Y and Rabin T, On the security of joint signature and encryption, in Process of Eurocrypt, LNCS, pp.83-107, 2002.

[7] Y. Sun and H. Li, Efficient signcryption between TPKC and IDPKC and its multi-receiver construction, 53 (3), pp.557-566, 2010.

[8] Q. Huang, D.S. Wong and G. Yang, Heterogeneous signcryption with key privacy, Computer Journal, 54(4), pp.525-536, 2011.

[9] F.Guo, Y.Mu and Zhide, Identity-based online/offline encryption, Lecture Notes in Computer Science 5143, pp.247-261, 2008.

[10] X. Huang, W. Susilo, Y. Mu and F. Zhang, On the Security of Certificateless Signature Schemes from ASIACRYPT 2003, Cryptology and Network Security (CANS), LNCS 3810:pp.132-5, 2005.

[11] F.Li and P. Xiong, Practical secure communication for integrating wireless sensor networks into the internet of things. IEEE Sensors Journal, 13(10), pp.3677–3684, 2013.

[12] J. K. Liu and J. Zhou, An efficient identity-based online/offine encryption scheme, ACNS 5536 of Lecture Notes in Computer Science, pp. 156-167, 2009.

[13] X. Boyen, Multipurpose Identity-Based Signcryption (A Swiss Army Knife for Identity-Based Cryptography). CRYPTO 2003, LNCS 2729, pp. 383-399, 2003.

[14] B. Libert and J.J. Quisquater, New Identity Based Signcryption Schemes from Pairings. IEEE Information Theory Workshop, pp. 155-158, 2003.

[15] P. Barreto, B. Libert, N. McCullagh and J. Quisquater. Efficient and provabley- secure identity-based signature and signcryption from bilinear maps. In ASIACRYPT, LNCS 3788, pp. 515-532, 2005.

[16] L. Chen and J. Malone-Lee. Improved Identity-Based Signcryption. In Proc. PKC, LNCS 3386, pp. 362-379, 2005.

[17] Z. Liu, Y.Hu, X. Zhang and H. Ma. Certificateless Signcryption Scheme in the Standard Model. Information Sciences, 2009.

[18] S. Miao, F. Zhang, S. Li and Y. Mu. On Security of a Certificateless Signcryption Scheme, 2010.

[19] P. LI, M.HE, X. LI and W. LIU, Efficient and Provably Secure certificateless Signcryption from Bilinear Pairings, JCIS 6:11, pp.3643-3650, 2010.

[20] L. Cheng and Q. Wen, An, Improved Certificateless Signcryption in the Standard Model. International Journal of Network Security.

[21] G.Yu, H.Yang, S. Fan,Y.Shen and W. Han, Efficient Certificateless Signcryption Scheme 2010, ISBN 978-952-5726-11-4, (ISECS 10) Guangzhou, P. R. China, 29-31, pp. 055-059, 2010.

[22] F. Zhang, A New Provably Secure Certificateless Signature. Proceedings of ICC08, pp.1685-1689, 2008.

[23] R. Guo, Q.Wen, H.Shi, Z.Jin, and Hua Zhang, Certificateless Public Key Encryption Scheme with Hybrid Problems and Its Application to Internet of Things. Hindawi Publishing Corporation Mathematical Problems in Engineering, 2014

[24] A. Perrig, J. Stankovic, and D. Wagner, Security in wireless sensor networks, communication. ACM,47(6), pp. 53-57, 2004.

[25] J. K. Liu, J. Baek, and J. Zhou. Online/offine identity-based signcryption re-visited. Cryptology ePrint Archive, 2010.