# A Signcryption Scheme from Certificateless to Identity-based Environment for WSNs into IoT

Benjamin Klugah-Brown
School of Computer Science and Engineering
University of Electronic Science and Technology of China,
611731, Chengdu, China

John Bosco Aristotle Kanpogninge Ansuura
School of Computer Science and Engineering
University of Electronic Science and Technology of China,
611731, Chengdu, China

Xia Qi
Big Data Research Center
University of Electronic Science and Technology of China 2nd
611731, Chengdu, China

## ABSTRACT
Wireless sensor network (WNS) is now an inevitable component of the internet of things (IoT), this integration creates new security challenges that exist between the sensor nodes and the internet host, thus, issue regarding setting up a non-compromised channel between these two ends. In this scheme we required that the sender of the message belongs to the internet host where huge computation can be done without incurring any delays or computational problem while the receiver belongs to the sensor node. The scheme is shown to be suitable and secure using random oracle of bilinear Diffie-Hellman assumption hence providing strong security for wireless sensors into internet of things.

## Keywords
Wireless sensor network, Internet of things, security, Certificateless signcryption, identity-based cryptography.

## 1. INTRODUCTION
The concept of internet of things (IoT) in modern times cannot be ignored as it has become imperative in daily life and it is indeed considered by both the academia and industry as the next frontier in future information technology and internet since it was proposed by Kelvin Ashton in 1991. This standard considered the idea that, all objects including human beings could be well managed and inventoried by computers if they are equipped with identifiers. It can be realized that the underlying idea is communication (usually through wireless medium) in which these objects such as radio-frequency identifiers (RFID), actuator, tags sensors, mobile phones, PDA's etc can interact. These applications and their complexity (intelligence) could interact and work together to achieve a common purpose. As stated earlier the integration of wireless sensor networks into the Internet of Things poses a new threat as far as security during communication is concern, and it's being investigated by many researchers today.

The environment is becoming more and more smart with technological advancement, hence it reliance on sensory data gathered from everyday life. The use of wireless sensor networks (WSNs) is inevitable in this regard. The WSNs are made up of several nodes connected to each sensors. But the main issues as at now is that these sensor nodes processing unit have limited computational power and inadequate capacity, hence the use of base station which is a powerful trusted devise that serves as an interface between the user and the nodes. Its applications include a variety of things including military sensing and tracking, environmental monitoring, target tracking, healthcare monitoring etc. Furthermore, today there are many views and perspectives that see the growing demand for the integration of WSNs into the Internet of Things(IoT), this need seem to arise from the heavily reliance on low resourced devices such as mobile phone, PDAs, etc. A user of the WSNs can read the data received from the sensors through the base station. If we hope to read the data anywhere in the world, we need to integrate the WSNs into the Internet as part of the IoT. However, new security challenges will appear, such as setup of a secure channel between a sensor node and an Internet host that supports end-to-end authentication and confidentiality services [1]. Note that the computational power and storage of a sensor node are limited. But an Internet host has strong computational power and storage. So we hope to design a secure communication scheme that fits such a characteristic.

To support the authenticity of public keys in the public key cryptography, there are three main infrastructures called public key infrastructure (PKI) identity-based cryptography (IBC) and Certificateless Public key cryptography (CLPKC). In the PKI, a certificate authority (CA) issues a certificate which provides an unforgeable and trusted link between the public key and the identity of a user by the signature of the CA. However, as well known apparent drawback inherent in both the PKI and IBC. That is;

With Public Key infrastructure (PKI), there is the need to manage certificates, including revocation, storage, and distribution. Also, verifying the validity of certificates before using them poses a huge problem. Nevertheless, the PKI technique has been widely developed and applied in the Internet despite the above mentioned problems.

The second issue has to do with managing the key escrow problem associated with IBC. As it has been well established, the IBC requires that a user's public key is derived directly from its identity information, such as telephone numbers, email addresses, and IP addresses. In this scheme the secret keys are generated for users by a trusted third party called private key generator (PKG), in which the public key are expected to be explicitly authenticity and verified without requiring any certificate. This characteristic requirement of the IBC offers a huge advantage over the traditional PKI as it provides the elimination for the need of certificates and its associated problem.

However, as stated above, the reliance on the PKG who generates all users' secret keys certainly causes the key escrow problem in the IBC as the PKG is suspected to likely misuse it privileges. But, for the WSNs, IBC is the best choice because there is no certificates problem. However, IBC is only suitable for small networks. For the Internet security, we need PKI technique. It is very important to note that despite the above mentioned shortcomings of PKI and IBC, it is still being used especially PKI which is currently being used in many application. Many researchers and firms have also started the application of the hybrid systems which involves the use of the combination of PKI and IBC.

## 1.1 Certificateless public key cryptography

Notwithstanding the challenges inherent in traditional PKI and IBC as we have seen, some researcher came out with a more efficient way to curb the note challenges. Al-Riyami and Paterson, 2003 introduced the concept of certificateless cryptography [2] which is very similar to IBC as there is no need for certificate but does not also rely on the so-called Trusted Third Party (TTP) that does surfer from the key escrow problem seen in IBC.

In certificateless cryptography there is a kind of TTP 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 does provide the user with a *partial private key* which it computes from the users identity and a master key. It is imperative to note that, the KGC must deliver the said partial private keys to the user in a secure way so as to achieve confidentially and also be authentic, the identity could also be any arbitrary string.

## 1.2 Motivation

Our motivation is derived from the fact that a security problem arises when there is a communication between a wireless sensor node and an Internet host, and the aim is to design a scheme between these two ends that provides a secure channel through which communication can take place by employing Certificateless signcryption methods. This scheme is expected per our design to supports end-to-end confidentiality, integrity, authentication, and nonrepudiation services. We require however, that the certificateless environment is used in the internet host and the IBC is use in sensor node for sending message and receiving message respectively. We derive our inspiration from the work of Li and Xiong in which they designed an online/offline signcryption scheme [1] which allowed a secure communication between a sender being in IBC (WSNs or Node) and a receiver being in PKI (internet or Host). However, in this paper we propose that sender belonging to the internet host (certificateless environment) and receiver belonging to the Wireless Sensor Node-WSNs (IBC). Our scheme employed certificateless signcryption (CLSC) introduced by Barbosa and Farshim in 2008 [3] and also in 2010 LI et al, [5] produce a scheme based on Barbosa and Farshim. Signcryption is a cryptographic primitive proposed by Zheng [6], which provides signature and encryption simultaneously, and has lower computational cost and communication overhead than the signature-then-encryption approach. A proper signcryption scheme should provide confidentiality and authenticity. Zhang [7] proposed another signcryption scheme based on elliptic curve, which saves about 58 *percent* computational cost and saving about 40 *percent* communication cost based on elliptic curve.

## 1.3 Related Work

In 2013 Li and Xiong proposed an online/offline signcryption scheme [1] to secure communication between a sensor node and an internet host for wireless network sensor into internet of things. Their scheme provides security solution for integration WSN into the IoT. Their scheme heavily relies on bilinear pairing solution for signcryption, which is a very good method for an online/offline scheme since it achieves signature and encryption in a single logical step. Identity-based signcryption was introduced by Molene-Lee [4].In 2008, Barbosa and Farshim first introduced the certificateless signcryption (CLSC) and proposed a CLSC scheme [3], which requires six pairing operations in the signcryption and unsigncryption phase. Recently, a proved security certificateless signcryption scheme in the standard model was proposed by Liu et al [7] which requires five pairing operations and one exponentiation in signcrypt and designcrypt phase. LI et al [5] produce a scheme based on Barbosa and Farshim. Our choice of this technique was as a result of the above stated problem inherent in both PKI and IBC [8]based schemes and also the efficient introduced by [3] we also require that the computational cost of sensor nodes and the internet is low. Recently, a number of efficient CLSC schemes [9, 10] have been proposed in

Certificateless cryptography including those done in a standard model [11].

## 1.4 Organization

The rest of the paper are organized as follows. We review some Preliminaries and security notion in Section 2. It is followed by our proposed scheme in Section 3. We analyze the performance of our scheme in Section 4. Our paper is concluded in Section 5.

# 2. PRELIMINARIES

We introduce the concept of bilinear map and some complexity assumption on which our scheme relies on.

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

(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 $G_2$.

## 2.1 Formal Model of Certificateless to IBSC

The scheme involves two parts, as shown in figure 1, namely; Certificateless signcryption that employs six polynomial time steps of algorithm and Identity-based signcryption conditions for the above mentioned cryptographic primitive signcryption schemes.

(1) **Setup** ($1^k$). Given a security parameter $1^k$, the KGC and PKG 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 KGC and PKG 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 which is use to produce user's secret value $\gamma_{ID}$ by taking user's identity $ID$ and system parameter $params$ as input and also returns a public key $PK$. The user obtained $\gamma_{ID}$ and $PK$ is used to construct a full private key.

(4) **Set-Private-Key** ($D_{ID}, \gamma_{ID}, params$). 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 $\gamma_{ID}$.

(5) **Extract-key-IBC** ($ID$). Here the user submits its identity to the PKG who uses the master secret key $msk_2$ and user's $ID$ to generate the corresponding private key $SK_{ID}$ in a secure way.

The signcryption and unsigncryption algorithms are as follows:

(6). **SC** ($M$, $SK_{IDa}, IDa, PK_{IDa}, ID_b, PK_b, params$). This is the signcryption algorithm. On input of a message $m \in M$ ($params$), sender's full private key $SK_{IDa}$, identity $IDa$ and public key $PK_{IDa}$, the receiver's identity $ID_b$ and public key $PK_b$, the global parameters $params$ and possibly some randomness $\alpha \in R(params)$, this algorithm outputs a ciphertext $\tau \in C(params)$ or an error symbol $\perp$.

(7). **USC**($\tau, SK_{IDb}, ID_b, PK_{IDb}, ID_a, params$). This is a deterministic unsigncryption algorithm. On input of a ciphertext $\tau$, 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 $\perp$.

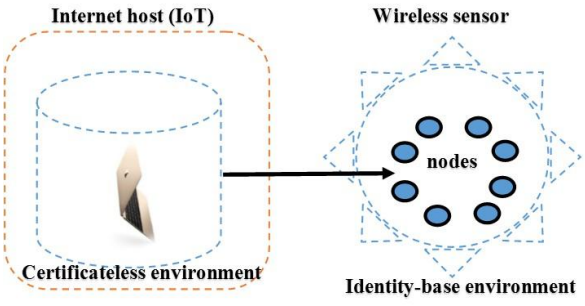


**Fig 1 general concept of integrating WSNs into IoT**

## 2.2 Security notion

As stated earlier, a signcryption scheme should be able to provide *confidentiality* and *authentication* as these form the bases 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

To capture confidentiality there are two games in which the adversary will interact with and the sender is $ID_A$ and receiver is $ID_b$. 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.

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 $msk_2$ to the adversary, while keeping master secret keys $msk_1$ 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** ($params, msk_1, ID$) 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 ($\tau_{ID}, PK_{ID}$). Finally, it sends the result of $SK_{ID}$ computed from **Set-Private-Key** ($\tau_{ID}$, $D_{ID}$) 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 ($\tau_{ID}, PK_{ID}$) = **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 $\tau$, 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** ($params, \tau, ID_a, PK_{IDa}, SK_{IDb}, ID_b, PK_{IDb}$), 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 selects $\mu \in_R \{0,1\}$, computes

$\tau^*$=**Signcrypt**
($params, m_\delta, SK_{IDa^*}$ , $ID_{a^*}$ , $PK_{IDa^*}$ , $ID_b$ , $PK_{IDb^*}$),and returns $\tau^*$ 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 $ID$, and a receiver's identity $ID$ on which it wishes to be challenged. $ID$ should not be queried to extract a private key in Phase 1. The challenger then selects $\mu \in_R \{0,1\}$ and gets.

$\tau^*$= **Signcrypt** $(params, m_\mu, SK_{ID_a^*}, ID_a, PK_{ID_a^*}, ID_b, PK_{ID_b^*})$, and returns $\tau^*$ 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 $\tau^*$ 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 a Type I adversary $F_I$ and a Type II 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 $msk_2, params$ 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 $(params, msk_1, ID)$ and computes the corresponding private key $SK_{ID}$ which it gives to $F_I$.

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

(3) Signcryption query: The challenge algorithm gets $SK_{IDa}$ and computes $\tau$ as the signcryption value when $F_I$ selects a message $m$, gets $ID_a$ and $ID_b$. The challenger sends $\tau$ to adversary $F$.

(4) Unsigncryption query: In this query algorithm $F$ selects a signcryption value $\tau$, 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 $(m^*, \tau^*, ID_a^*, ID_b^*)$. 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, $\tau^*$ was not returned by the signcryption oracle on the input $(m^*, \tau^*, ID_a^*, ID_b^*)$ during the attack stage. $F$ wins the game if;

**Unsigncrypt** $(params, \tau^*, SK_{ID_a^*}, ID_a, PK_{ID_a^*}, ID_b^*, PK_{ID_b^*})$ 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 $ID$ in the above definition. Again, this condition corresponds to the stringent requirement of insider security for signcryption.

## 3. OUR PROPOSED SCHEME

In this section, we propose an efficient certificateless communication scheme which is based on certificateless signcryption scheme. But our scheme requires that the receiver belongs to IBC which means that during the unsigncryption stage there will be no need for partial keys and secrete values hence reducing the computational overhead. Our scheme requires that the sender in certificateless environment to send a message to a receiver in IBC. Here we will denote the sender who is in the internet host as Alice(a),while the receiver Bob(b) in the sensor node.

### 3.1 Scheme Description

We now present our certificateless-identity-based signcryption scheme which can be seen as an Encrypt-then-Sign construction where randomness is shared between signature and encryption schemes. Our scheme consist of five algorithms **Setup:**

We choose four cryptographic hash functions:

$H_1 : \{0,1\}^* \rightarrow G_1$

$H_2 : \{0,1\}^* \rightarrow \{0,1\}^k$

$H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_P^*$

$H_4 : \{0,1\}^* \rightarrow \mathbb{Z}_P^*$

Since the communication is between certificateless and identity-based environments,we prefer that the KGC and the PKG Select different master secret values, such that the KGC selects $s_1 \in_R \mathbb{Z}_P^*$ and PKG selects $s_2 \in_R \mathbb{Z}_P^*$ , while setting $mpk = s_1 P$ and $P_{pub} = s_2 P$ respectively. The KGC inputs $(ID, msk)$ and using partial secret key extraction algorithm returns $D_{ID} = sH(ID)$. The user then selects a secret value $\tau \in_R \mathbb{Z}_P^*$ and computes $PK = \tau P$ as the public key and sets $SK = (\tau, D_{ID})$ as the full private key.

The KGC choose a random value $s \in_R \mathbb{Z}_P^*$. Let $P_{pub} = s_2 P$ ($P$ is a random generator of $G_1$)be the known key of the system. The following represent the public parameter $params; < G_1, G_2, p, e, P_{pub}, g, H_1, H_2, H_3, H_4 >$.

**Extract-Partial-Private-Key:** a user in the certificateless environment submits $ID$ as identity to an algorithm run by KGC which take as input $msk, params$, while $ID \in \{0,1\}^*$. The KGC set $D_{ID}$ and returns the partial private key as

$D_{ID} = s_1 H(ID)$ 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 $\tau_{ID}$which can be used to construct both the public and the private keys of the user.

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

**Extract-key-IBC** $(ID)$**:** Here the user submits it identity to the PKG who uses the extract algorithm to generate the corresponding private key $SK_{IDb}$.

The following algorithm describes the signcryption and unsigncryption stages:

**Signcryption:**

(1) $\alpha \in_R \mathbb{Z}_p^*$, compute $U = \alpha P, T = \hat{e}(P_{pub}, Q_b)^{\alpha}$

(2) set $h = H_2(U, T, ID_b)$

(3) compute $C = m \oplus h$

(4) set $\Upsilon = H_3(U, C, ID_a, PK_a)$

(5) compute $G = D_{ID} + \tau \Upsilon$

(6) Return $\delta = (U, C, G)$ as the ciphertext.

**Unsigncryption:**

(1) compute $\delta = (U, C, G)$

(2) Set $\Upsilon = H_3(U, C, ID_a, PK_a)$

(3) check if $\hat{e}(P_{pub}, Q_a)\hat{e}(UPK_a, \Upsilon) \neq \hat{e}(P, G)^U$ return $\perp$

(4) $T = \hat{e}(SK_{ID}, U)$

(5) $h = H_2(U, T, ID_b)$

(6) $m = C \oplus h$

(7) Return $m$

The figure 2 below shows the full scheme between Alice in the internet host sending message to Bob in sensor node.

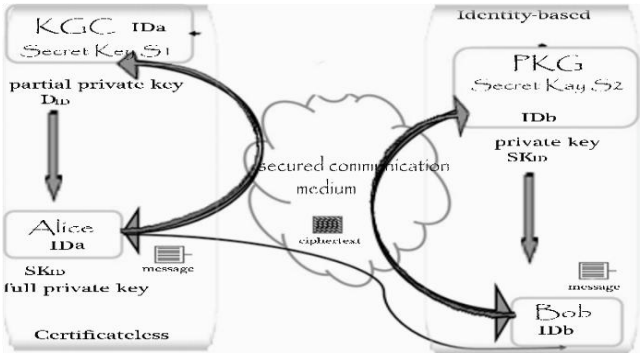


**Figure 2. Procedure for secure communication**

## 4. ANALYSIS OF THE SCHEME

In this section we present the consistency, including correctness, security and performance.

Correctness

The correctness can be easily verified: $\hat{e}(P_{pub}, Q_a)\hat{e}(UPK_a, \Upsilon) = \hat{e}(P, G)^U$. Bob can then verify the signature as follows;

$= \hat{e}\ (s_1 P, Q_a)\ \hat{e}\ (\tau P, U\Upsilon)$

$= \hat{e}\ (s_1 P, Q_a)\hat{e}(P, U\tau\Upsilon)$

$= \hat{e}\ (P,\ s_1 Q_a)\ \hat{e}(P, U\tau\theta)$

$= \hat{e}\ (P, (s_1 Q_a\ +\ U\tau\Upsilon))$

$= \hat{e}\ (P, (D_{ID}\ +\ U\tau\Upsilon))\ =\ \hat{e}(P, G)^U.$

Correctness for decryption

We give the correctness of decryption as follows;

$\hat{e}(U, SK_{ID}) = \hat{e}(P_{pub}, Q_b)^{\alpha}$

$= \hat{e}\left(P_{pub}, Q_{ID_B}\right)^{\alpha}$

$= \hat{e}(s_2 P, Q_b)^{\alpha}$

$= \hat{e}(P, sQ_{ID_B})^{\alpha}$

$= \hat{e}(\alpha P, sQ_b)$

$= \hat{e}(\alpha P, SK_{ID})$

$= \hat{e}(U, SK_{ID})$

The analysis of this scheme relies on the security proofs presented in [6] with a variation since the unsigncryption 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 $mpk_1 = cP$ 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 $M_1$ while $Q_{ID}$ is returned. If $v = \ell$, the challenger produces $Q_{ID} = bP$ and adds $(v, ID, \perp)$ to $M_1$.

**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$M$ containing a tuple $(ID, PK, \perp)$ upon an input of $ID$and$PK$. If $v \neq \ell$, the challenger searches $M_1$ for entry $(ID, PK, \tau)$, it then proceeds to get new key pair and returns $(\tau, \alpha cP)$ 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 $\vartheta = \tau U$;

(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 $T = \hat{e}(SK_{ID}, mpk_2)$ 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 $M_2$, 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 $M_2$ 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 $M_i$ containing the pair $(PK_{IDb}, ID_a)$ 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.

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 $M_2$, also $M_1$ 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 keeps $msk_1$. 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 $\tau \in_R \mathbb{Z}_p$ generates a new key pair $(\tau, PK)$, if $v \neq \ell$ on the $v^{th}$ non-repeated query. $L_2$ is updated with the tuple $(v, ID, \tau, PK)$. 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 $(v, ID, PK, \tau)$ by running the request public key on every new query on $ID$. The challenger proceeds so long as $v \neq \ell$ to return $(\tau, msk_1 \alpha P)$ 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 $M_3$ 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 $M_2$ for the entry tuple $(U, T, ID, r)$.

(3) The challenger returns $r$ and updates an empty list $M_2$ 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 $M_2$ 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 $M_2$, 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 $M_2$. 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 $(\omega, ID_b^*, PK^*, \tau)$ 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 $M_2$, also $M_1$ has at most $q_T$ elements with probability $\frac{1}{q_T}$. However, the adversary may not be able to identify the right signcrypted message unless it runs a query on $H_2$ containing tuple$(U^*, T^*, \vartheta^*, ID_\ell, PK^*)$ from the list $M_i$. 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 functions are modelled as random oracles. More precisely, there exists an algorithm B which uses A to solve the GDH problem such that:*

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

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 B recovers $M_i$ 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 $M_3$ 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 $M_2$ $(U, \vartheta, ID, PK, r)$ 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 $M_2$ 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 $M_2$ 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 $H_3$.

In this section we compare the efficiency in the proposed scheme [4]. Note the abbreviations used in Table 1: 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).

| Scheme | Security | | Performance | | |
|---|---|---|---|---|---|
| | **CCA2** | **CMA** | **Mul** | **Exp** | $\hat{e}$ |
| J. Malone- Lee[4] | No | Yes | 3 | 1 | 6 |
| Our Scheme | Yes | Yes | 3 | 2 | 4 |

**Table 1 Security and performance comparison**

Table 1 shows that our scheme relatively efficient that [4] since pairing operations are much expensive than exponentiation operation. Also our scheme achieves the major requirements for any signcryption scheme that is both CCA2 and CMA.

# 5. CONCLUSION

Our proposed certificateless to identity-based signcryption scheme which requires a user belong to a certificateless environment to send a message to a receiver in identity-based environment provides a secured channel between these two ends thereby supporting confidentiality, integrity, authentication and non-repudiation service as expected of every security scheme. In this scheme the sender need not worry about computational cost since it does not need any certificate as inherent in PKI hence huge computations can be done. Our scheme also makes use of only one pairing operation during signcryption hence suitable for resource constraint devices. However the receiver computations are reduced since it need not compute private keys by itself. However, we would like to continue this research by using heterogeneous online/offline techniques as it produce a better lower computational cost and also appropriate for WSNs.

# 6. ACKNOWLEDGMENTS

This paper is supported by the Big Data Research center, University of Electronic Science and Technology of China.

# 7. REFERENCES

[1] 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

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

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

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

[5] 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

[6] Y.Zheng,H.Imai. *How to Construct Efficient Signcryption Schemes on Elliptic Curves*. Information Processing Letters, 68(5): pages 227-233, 1998

[7] F. Zhang. A New Provably Secure Certificateless Signature.proceedings of ICC08, pages 1685-1689,2008

[8] D.Boneh and M. Franklin. *Identity-Based Encryption,SIAM J. of Computing, Vol. 32, No. 3, pp. 586-615, 2003. Crypto* 2001, volume 2139 of Lecture Notes in *Computer Science, pages 213229, Springer-Verlag*, 2001

[9] F. Li, M. Shirase, T. Takagi, "Certificateless hybrid signcryption", *Mathematical and Computer Modelling*, vol.57, pp. 324-343, 2013.

[10] F. Monrose, M. Reiter, Q. Li, et al, "Towards voice generated cryptographic keys on resource constrained device", in: *Proceedings of the 11th USENIX Security Symposium*, San Francisco, CA, USA, pp. 1-14, 2002.

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