Lightweight certificateless and provably-secure signcryptosystem for the internet of things

Kim Thuat Nguyen, Nouha Oualha, Maryline Laurent

Lightweight Certificateless and Provably-Secure Signcryptosystem for the Internet of Things

Kim Thuat Nguyen, Nouha Oualha
CEA, LIST, Communicating Systems Laboratory
91191 Gif-sur-Yvette CEDEX, France
Email: kimthuat.nguyen@cea.fr, nouha.oualha@cea.fr

Maryline Laurent
Institut Mines-Telecom, Telecom SudParis
UMR CNRS 5157 SAMOVAR
9 rue Charles Fourier, 91011 Evry, France
Email: maryline.laurent@telecom-sudparis.eu

***Abstract*—In this paper, we propose an elliptic curve-based signcryption scheme derived from the standardized signature KCDSA (Korean Certificate-based Digital Signature Algorithm) in the context of the Internet of Things. Our solution has several advantages. First, the scheme is provably secure in the random oracle model. Second, it provides the following security properties: outsider/insider confidentiality and unforgeability; non-repudiation and public verifiability, while being efficient in terms of communication and computation costs. Third, the scheme offers the certificateless feature, so certificates are not needed to verify the user's public keys. For illustration, we conducted experimental evaluation based on a sensor Wismote platform and compared the performance of the proposed scheme to concurrent schemes.**

***Keywords*—signcryption, security, Internet of Things, random oracle model proof**

I. Introduction

The Internet of Things (IoT) is about the interconnection of devices to the Internet, including smart objects and embedded computing devices, such as: sensors, actuators, RFID tags, smartphones or even our everyday life friendly devices (e.g. thermostats, fridges, ovens, washing machines and TV). The more the IoT devices are deployed, the higher is the need for security concerns. Indeed, the IoT are generally considered as very rich sources of data. If not protected, these data can be abused to spy on our activities and consequently threaten our security and privacy.

The energy consumption is a critical criteria for any security protocol and mechanism deployed on low-cost computing platforms used in IoT. Usually, security solutions require expensive cryptographic operations, which consume rapidly the energy available on resource-limited devices and therefore reduce the life time dedicated to application.

Symmetric approaches can establish secure communications between parties with no complexity computations required. If the symmetric key is shared among all the devices, the system security is weakened. While, if each pair of communicating devices has its own pre-shared symmetric key, the key bootstrapping mechanism becomes more difficult to manage, especially in the context of IoT where the number of connected devices composing the network is generally large. Asymmetric approaches for secure communication establishment is a good alternative since they are able to establish a secure communication between two entities who do not share any common keys. In such setting, it is important to authenticate the public keys to avoid spoofing and masquerading attacks. Generally, the validity of public keys is provided by means of certificates. However, the verification and management of certificates require important computation operations, bandwidth for communicating with remote entities and sophisticated revocation mechanisms. The aforementioned requirements are not appreciated for low-bandwidth and resource-constrained environments.

In order to guarantee the unforgeability, integrity and confidentiality of communications, one recommended approach is to sign then encrypt using public key encryption. This approach is proved to be much more costly in terms of computation and communication complexity than the signcryption approach. This latter, initially proposed in [3], combines simultaneously signature and encryption. To alleviate signcryption schemes from the issue of public key authenticity verification, Barbosa et al. [26] proposes the notion of certificateless signcryption. The proposed cryptographic primitive inherits the properties of certificaless cryptography [29] which integrates the identity-based cryptography (IBC) [30] with the public key cryptography (PKC) but removes both key-escrow property in IBC and certificates in traditional PKC. Several certificateless signcryption schemes have been proposed in the literature [26], [27], [25], [24]. However, they still require multiple modular exponentiations and pairing-based operations, which may be not practical for resource-constrained devices in the context of IoT.

Our contribution: In this paper, we first propose a new elliptic curve based signcryption scheme derived from KCDSA [2], which satisfies strong security properties in the random oracle model [36]: confidentiality against outsider chosen-ciphertext attacks, unforgeability against insider chosen-message attacks. Second, we prove the security of the proposed scheme via a sequence of games. Third, at the extra cost of one extra point multiplication, our proposal can achieve the insider confidentiality property. Fourth, we show that our scheme removes the need for certificates and still presents the best performance in comparison with related work. Finally, we present experimental performance results of known existing signcryption schemes based on an emulated sensor Wismote platform and demonstrate the efficiency of our proposed scheme.

II. Preliminaries

In this section, we review security assumptions used in our work and define the threat model for the signcryption schemes.

A. Abbreviations and Definitions

The terms and definitions used throughout the rest of this paper are presented as follows:

- $P+Q$ denotes the addition of two elliptic curve points P and Q.
- $[t]P$ denotes the addition of P with itself t times.
- $s||t$ denotes the concatenation of two strings s and t.
- $x \xleftarrow{\$} \mathbb{X}$ denotes the operation of assigning to x a randomly chosen element of $\mathbb{X}$.
- $\perp$ denotes the error symbol.
- $l_p(k) : \mathbb{N} \rightarrow \mathbb{N}$ is a function determining the length of p, given a security parameter k.
- A PPT adversary denotes a probabilistic polynomial time adversary.

B. Security Assumptions

1) Computational primitives: Let $\mathbb{G}$ be a cyclic group of prime order p. For our purposes, $\mathbb{G}$ is a subgroup of points of a suitable elliptic curve $E(\mathbb{F}_p)$ over finite field. P is a generator of $\mathbb{G}$. We define the following security assumptions:

Definition 1 (Decisional Diffie-Hellman (DDH) Problem)**.** *Given the "Diffie-Hellman tuples"* $< P, [a]P, [b]P, [c]P >$*, decide whether* $ab \equiv c \,(mod\, p)$ *or not.*

Definition 2 (Computational Diffie-Hellman (CDH) Problem)**.** *Given* $< P, [a]P, [b]P >$*, for unknown* $a, b \in \mathbb{Z}_p$*, compute* $[ab]P$*.*

Definition 3 (Gap Diffie-Hellman (GDH) Problem)**.** *Given that the DDH problem is easy in* $\mathbb{G}$*, solve an instance of the CDH problem* $< P, [a]P, [b]P >$*.*

Definition 4 (Discrete Logarithm Problem (DLP))**.** *Given the two points P and Q on the elliptic curve over finite field* $\mathbb{F}_p$*, find* $d \in \mathbb{Z}_p$ *so that* $[d]P = Q$*.*

Definition 5 (Gap Discrete Log (GDL) Problem)**.** *Given that the DDH problem is easy in* $\mathbb{G}$*, solve an instance of the DLP problem* $< P, Q >$*.*

2) A signcryption scheme: We define a signcryption scheme as a tuple of four PPT algorithms (Setup, KeyGen, Signcrypt, Unsigncrypt) with the following functionalities:

– $\mathsf{Setup}(k) \rightarrow cp$. Given a security level parameter k, output the public parameters cp. The other functions takes cp as an implicit input.

– $\mathsf{KeyGen}(cp) \rightarrow (sk_S, pk_S), (sk_R, pk_R)$. Generate public/private pair of keys for two parties (Sender and Receiver).

– $\mathsf{Signcrypt}(sk_S, pk_S, pk_R, m) \rightarrow C$ or $\perp$. Given the public/secret keys of the Sender, the public key of the Receiver and a message m, return either a signcryptext C or $\perp$.

– $\mathsf{Unsigncrypt}(pk_S, sk_R, pk_R, C) \rightarrow m$. Given the signcryptext C, the public/secret keys of the Receiver, the public key of the Sender, return either a message m or $\perp$.

3) One-time symmetric encryption: As earlier given in [6] and [7], we define the one-time indistinguishability (OT-IND) property of the symmetric key encryption (SKE).

Definition 6 (OT-IND for symmetric encryption scheme)**.** *Let* $SKE = (Enc, Dec)$ *be a bijective one-time symmetric encryption scheme with security parameter k,* $\mathcal{A}$ *be a PPT adversary against the security of* SKE *in the sense of OT-IND. The advantage of* $\mathcal{A}$ *to win the following game must be negligible:*

– The challenger uniformly chooses at random a secret $K \in \{0,1\}^l$*, where l is an integer calculated from k*

– $\mathcal{A}$ *is given the security parameter k. It then outputs a pair of messages* (m_0, m_1) *of equal length and passes them to the challenger.*

– On receiving this pair, the challenger selects a bit $b \xleftarrow{\$} \{0,1\}$ *and outputs the ciphertext* $C = Enc(K, m_b)$ *or* $\perp$ *if the messages do not have equal length.*

– $\mathcal{A}$ *receives the ciphertext C and outputs* b'*.* $\mathcal{A}$ *wins the game if* $b' = b$*.*

$\mathcal{A}$*'s advantage is defined to be* $Adv_{\mathcal{A}}^{OT-IND}(k) = 2Pr[b' = b] - 1$*.*

C. Security models for signcryption schemes

This section presents the security models for two security notions of signcryption schemes: confidentiality against chosen-ciphertext attacks (CCA), which is also known as semantic security, and the unforgeability against chosen-message attacks (CMA). We consider a multi-user setting as already studied in [6], [9]. Concisely, there exist many other users in addition to the attacked Sender (S) and Receiver (R). The attacker can be either an insider or outsider that acts by replacing the sender/receiver public keys at will when accessing the signcryption/unsigncryption oracles. In the outsider setting, an attack is perpetrated by a third party which is different from S and R. On the other hand, an attack in the insider setting is issued from an internal party, meaning that the attacker is a compromised S or R. In such model, the owner of a private key is unable to retrieve any information on a ciphertext previously signcrypted by himself without knowing the randomness used to produce that ciphertext. Thereafter, this paper refers to confidentiality as the confidentiality against CCA in the outsider model, and it refers to unforgeability as the unforgeability against CMA in the insider model.

Definition 7 (SC-IND-CCA2 [6])**.** *Let* $\mathcal{A}$ *be a PPT adversary against the confidentiality of a signcryption scheme between the (fixed) sender S, and the (fixed) receiver R, with security parameter k.* $\mathcal{A}$ *has negligible advantage to win the following game, denoted as* $EXPT_{\mathcal{A}}^{SC-IND-CCA2}(k)$*:*

– The challenger runs the algorithms Setup *and* KeyGen *to generate keying material for S and R.* (sk_S, sk_R) *are kept secret while* (pk_S, pk_R) *are given to* $\mathcal{A}$*.*

– $\mathcal{A}$ *can make calls to the signcryption and unsigncryption oracles. On each signcryption query,* $\mathcal{A}$ *produces a pair* (m, pk_B) *at will where* pk_B *is an arbitrary receiver's public key (that public key may differ from* pk_R*) and* m *is the message. On receiving this pair, the signcryption oracle* $\mathcal{O}_{SC}$ *returns the result of* $\mathsf{Signcrypt}(sk_S, pk_S, pk_B, m)$ *to* $\mathcal{A}$*. On each unsigncryption query,* $\mathcal{A}$ *produces a pair* (pk_A, C) *at will where* pk_A *is an arbitrary sender's public key and C is a signcryptext. On receiving this pair, the unsigncryption oracle* $\mathcal{O}_{USC}$ *returns the result of* $\mathsf{Unsigncrypt}(pk_A, sk_R, pk_R, C)$ *to* $\mathcal{A}$*.*

– $\mathcal{A}$ *outputs a pair of messages of equal length* (m_0, m_1)*. On receiving this pair, the challenger selects a bit* $b \xleftarrow{\$} \{0,1\}$ *and sends the challenge ciphertext* $C_{RS} = \mathsf{Signcrypt}(sk_S, pk_S, pk_R, m_b)$ *to* $\mathcal{A}$*.*

– $\mathcal{A}$ *submits a number of queries to* $\mathcal{O}_{SC}$ *and* $\mathcal{O}_{USC}$ *as* $\mathcal{A}$ *did in previous steps. However, it is not allowed to query* O_{USC} *on* (pk_S, C_{RS})*. Note that* $\mathcal{A}$ *can query* $\mathcal{O}_{USC}$ *on* (pk_A, C_{RS}) *for any* $pk_A \neq pk_S$ *and query* $\mathcal{O}_{USC}$ *on* (pk_S, C) *for any* $C \neq C_{RS}$*.*

– At the end of the game, $\mathcal{A}$ *outputs* b' *and wins the game if* $b' = b$*.*

$\mathcal{A}$*'s advantage is defined to be* $Adv_{\mathcal{A}}^{SC-IND-CCA2}(k) = 2Pr[b' = b] - 1$*.*

Definition 8 (SC-UF-CMA [6])**.** *Let* $\mathcal{A}$ *be a PPT adversary against the unforgeability of a signcryption scheme with se-*

curity parameter k. $\mathcal{A}$ has negligible advantage to win the following game, denoted as $EXPT_{\mathcal{A}}^{SC-UF-CMA}$:
– The challenger runs the algorithms Setup *and* KeyGen *to generate a pair of public/private keys (sk_S, pk_S) for the sender S.*
– $\mathcal{A}$ can make calls to $\mathcal{O}_{SC}$, but not to $\mathcal{O}_{USC}$, because it can generate by itself a pair of receiver's private/public keys. On each signcryption query, $\mathcal{A}$ produces a pair (m, pk_B) at will where pk_B is an arbitrary receiver's public key and m is the message. On receiving this pair, $\mathcal{O}_{SC}$ returns the result of $\mathsf{Signcrypt}(sk_S, pk_S, pk_B, m)$ *to $\mathcal{A}$.*
– At the end of the game, $\mathcal{A}$ outputs a pair of receiver's private/public keys (sk_R, pk_R) and a signcrypted text C_{RS}. We say that $\mathcal{A}$ wins the game if the following conditions are satisfied: (i) C_{RS} is a valid signcryptext from S to R (this means that the unsigncryption process is done under the sender's public key pk_S and the receiver's private key sk_R); (ii) $\mathcal{A}$ did not query on (m_{RS}, pk_R) to $\mathcal{O}_{SC}$, where m_{RS} is the plaintext of the signcryptext C_{RS}.

III. Our proposed sign-cryptosystem for IoT

In this section, we present a lightweight signcryption scheme based on the standardized signature KCDSA [2]. We start by describing the architecture of our solution. Then, we introduce our proposal in great detail. Finally, we show that our scheme is exempted from certification requirements.

A. Architecture

The considered scenario throughout this document contains the following actors:

- two parties: sender S and receiver R, that do not share any pre-established credentials.
- a Key Distribution Center (KDC), who provides the root of trust for both parties.

The KDC is in charge of providing key material for all communicating devices. In this document, it is considered that there is only one KDC. Applications may use multiple or distributed KDCs and hence may need different system parameters (general parameters, public/private keys). The mechanism for deciding which system parameters to use (when more than one KDC is available) is out of scope of this paper.

The KDC first selects a secret value mk as the system secret master key. The KDC's public key PK_{KDC} is then generated from mk. This public key is the root of trust for both parties. The KDC then provides key material for each device in the system. The idea of key construction is inspired from works in [1]. It defines a public validation token (PVT) to validate the relation between the secret signing key of each device and PK_{KDC}. Our approach uses PVT to cryptographically bound the device's public key to PK_{KDC}, instead of having a pair of public/private keys and a certificate. The PVT does not require any further explicit certification. KDC also attributes a short unambiguous identifier for each device. A device identification must be unique and can be renewed along with its key material by the KDC. Note that the transfer of key parameters to each device must be secure.

B. A new lightweight certificateless Diffie-Hellman based signcryption scheme

This section presents our lightweight signcryption scheme derived from the KCDSA signature scheme [2], but in the elliptic curve setting because of its efficiency in terms of computational cost [40]. We name our scheme SCKWC.

Setup: Depending on the security parameter k as input, the KDC selects an elliptic curve $E(\mathbb{F}_p)$ over finite field $\mathbb{F}_p$. $E(\mathbb{F}_p)$ is actually defined by the set of parameters (p, a, b, G, n, h), where p is a prime modulus, two elements a, b specify the elliptic curve $E(\mathbb{F}_p)$, G is the base point, n is the prime order of G and h is the cofactor. Further guidance on the selection of recommended domain parameters for elliptic curve cryptography can be found in [21]. The cryptographic hash functions are also chosen, such that $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$, $H_0 : \{0,1\}^* \rightarrow \{0,1\}^l$. From a chosen master key mk, the KDC computes $PK_{KDC} = [mk]G$. Each entity A is uniquely identified with ID_A. A's public/private key pair is generated by KDC as follows:

- Generate a public validation token $PVT_A = [x_A]G$, where x_A is a random number on $\mathbb{Z}_p^*$.
- Compute the private key for A: $priv_A = (mk + x_A.H_1(ID_A||PVT_A||G||PK_{KDC}))^{-1}$
- Compute the public key for A: $PK_A = [priv_A^{-1}]G$

Signcrypt: To signcrypt a message m intended to R, S executes the following steps:

1) Check the validity of R's public keys, as described in section III-C.
2) Choose randomly $x \xleftarrow{\$} \mathbb{Z}_p^*$.
3) Compute $K = [x]PK_R$.
4) Generate a secret key: $\tau = H_0(PK_S||PK_R||K)$.
5) Compute $r = H_1(PK_S||PK_R||K||m)$.
6) Compute $s = priv_S \cdot (x - r) \bmod p$.
7) Compute $c = Enc_\tau(m)$.
8) Send (r, s, c) to R

Unsigncrypt: Upon receiving the tuple (r, s, c), R has to perform the procedure as follows:

1) Check the validity of S's public key, as described in section III-C.
2) Compute $W = [s]PK_S + [r]G$.
3) Compute $K = [priv_R]^{-1}W$.
4) Get the secret keys: $\tau = H_0(PK_S||PK_R||K)$.
5) Compute $Dec_\tau(c) = m$.
6) Verify that $r = H_1(PK_S||PK_R||K||m)$.

Correctness: if a signcryptext (r, s, c) is generated by a legitimate sender, then the value of $[priv_R]^{-1}W = [priv_R]^{-1}.([(x-r).priv_S.priv_S^{-1}]G + [r]G) = [priv_R]^{-1}[x]G$ is equal to $[x]PK_R$, which means that $r = H_1(PK_S||PK_R||K||m)$.

C. Public key validation

This section describes the algorithm to be executed in the first step of signcryption and unsigncryption phases. Concretely, it explains the process of validating the public pair (PK_I, PVT_I) for any entity I. To validate these public values, the used algorithm requires the identification of I, namely ID_I and the KDC public key PK_{KDC}. The following checks must be passed successfully:

- Validate that PK_I and PVT_I lie in the same defined elliptic curve E.
- Compute $H_1(ID_I||PVT_I||G||PK_{KDC})$, as an integer number on $\mathbb{Z}_p$.
- Validate that $PK_I = PK_{KDC} + [H_1(ID_I||PVT_I||G||PK_{KDC})]PVT_I$.

The algorithm above can be only executed at the first run of the protocol. R and S may save the trusted public parameters of the other party for future uses. Besides, the revocation

of I's public values can be checked easily if the identifier ID_I is correctly generated. For instance, the identifier format can include a timestamp in order to automatically enable the expiration of key material. An example of how to create such identifier can be found in [23].

IV. Security analysis

In this section, we give a formal security analysis of our proposal. Our analysis is inspired from works conducted in [6] and [7]. First, we define several security notions needed for the proof. Then, we prove that the confidentiality and unforgeability of SCKWC are tightly related to the hardness of GDH and GDL problems.

A. Notation for the security proof

The security proof requires complex interactions between the oracles. Hence, we use two lists L_0 and L_1 to keep track of queries to and responses from the hash, signcryption and unsigncryption oracles. Precisely, L_0 contains the values of type $(PK_A, PK_B, W, K, \tau) \in \mathbb{G}^2 \times \mathbb{G}_\star^2 \times \{0,1\}^l$. Likewise, L_1 contains the values of type $(m, PK_A, PK_B, W, K, r) \in \{0,1\}^* \times \mathbb{G}^2 \times \mathbb{G}_\star^2 \times \mathbb{Z}_p$. For any set $\mathcal{X}$, we define $\mathcal{X}_\star = \mathcal{X} \cup \{\star\}$, where the symbol $\star$ denotes the parameter that can not be calculated by the simulation. We define $\mathcal{O}$ to be a DDH oracle that is able to determine whether or not the tuple $([a]P, [b]P, [c]P)$ satisfies $ab \equiv c(mod\, p)$. We index records in the list L_i by the set I_{L_i} $(i = 0, 1)$. The symbol ε defines an empty string. The symbol $\cdot$ specifies a parameter that "matches" any values. That is, if there exists $(x, y, \cdot) = (u, v, w)$ then we have $x = u$ and $y = v$. For any variable X calculated by a simulator, X^* is also a simulated value but its value is the same as the value calculated by the real oracles. We additionally consider that q_i (for $i = 0, 1$), q_{SC}, and q_{USC} are the maximum number of queries made to H_i, signcryption and unsigncryption oracles, respectively.

B. Confidentiality of our scheme

Theorem 9. *In the random oracle model, given a PPT adversary $\mathcal{A}$ against the SC-IND-CCA2 security of the SCKWC signcryption scheme, there exists a PPT adversary $\mathcal{B}_1$ against the GDH problem and a PPT adversary $\mathcal{B}_2$ against the OT-IND property of the symmetric encryption scheme such that:* $Adv_{\mathcal{A}}^{SC-IND-CCA2}(k) \leq 2Adv_{\mathcal{B}_1}^{GDH}(k) + Adv_{\mathcal{B}_2}^{OT-IND}(k) + \frac{2q_{SC}(q_1+q_{SC}+q_{USC})}{2^{l_p(k)}} + \frac{2(q_{SC}+q_{USC})}{2^{l_p(k)}}$.

Proof: We will prove the theorem via a sequence of games [35]. We denote S_i to be the event that $\mathcal{A}$ outputs the bit b' in game G_i and $b' = b$.

Game $\mathbf{G_0}$: This is the original attack game $EXP_{\mathcal{A}}^{SC-IND-CCA2}(k)$ defined in Definition 7. Hence,

$$Pr[S_0] = \tfrac{1}{2} + \tfrac{1}{2} Adv_{\mathcal{A}}^{SC-IND-CCA2}(k)$$

Game $\mathbf{G_1}$: This game replaces two random oracles H_0, H_1 by two random oracle simulators $\mathsf{H_0Sim}$ and $\mathsf{H_1Sim}$. We maintain the simulation of oracles by storing historical queries and responses into the two lists L_0 and L_1. We first define rules on how to determine membership in the list L_0 and L_1, as described in Figure 1. Based on these rules, we simulate $\mathsf{H_0Sim}$ and $\mathsf{H_1Sim}$ as denoted in Figure 2.

We observe that the simulation of H_0 and H_1 is modeled as random oracles and the consistency among hash queries is ensured by the lists L_0 and L_1. Besides, we assume in this game that the signcryption and unsigncryption oracles are perfect. As a result, we have that Game 1 is equivalent to Game 0. Thus,

$L_0Rule(PK_A, PK_B, K, W)$:
If $(PK_A, PK_B, \cdot, K, \cdot) = (PK_{Ai}, PK_{Bi}, W_i, K_i, \tau_i)$ or $(PK_A, PK_B, W, \cdot, \cdot) = (PK_{Ai}, PK_{Bi}, W_i, K_i, \tau_i)$, $i \in I_{L_0}$ then $\tau \xleftarrow{\$} \tau_i$
else if there exists $(PK_{Ai}, PK_{Bi}, W_i, K_i, \tau_i) \in L_0$ and $\mathcal{O}(W, PK_B, K_i) = 1$ or $\mathcal{O}(W_i, PK_B, K) = 1$ then $\tau \xleftarrow{\$} \tau_i$
else $\tau \leftarrow \perp$
return τ

$L_1Rule(m, PK_A, PK_B, K, W)$:
If $(m, PK_A, PK_B, \cdot, K, \cdot) = (m, PK_{Ai}, PK_{Bi}, W_i, K_i, r_i)$ or $(m, PK_A, PK_B, W, \cdot, \cdot) = (m, PK_{Ai}, PK_{Bi}, W_i, K_i, r_i)$, $i \in I_{L_1}$ then $r \xleftarrow{\$} r_i$
else if there exists $(m, PK_{Ai}, PK_{Bi}, W_i, K_i, \tau_i)$, $i \in I_{L_1}$ and $\mathcal{O}(W_i, PK_B, K) = 1$ or $\mathcal{O}(W, PK_B, K_i) = 1$ then $r \xleftarrow{\$} r_i$
else $r \leftarrow \perp$
return r

Fig. 1. Functions which determine membership in the list L_0 and L_1 from partial information

$\mathsf{H_0Sim}(PK_A, PK_B, K)$:
$\tau \leftarrow L_0Rule(PK_A, PK_B, K, null)$
If $\tau = \perp$ then $\tau \xleftarrow{\$} \{0,1\}^l$; $Add(PK_A; PK_B, \star, K, \tau)$ to L_0.
return τ
$\mathsf{H_1Sim}(m, PK_A, PK_B, K)$:
$r \leftarrow L_1Rule(m, PK_A, PK_B, K, null)$
If $r = \perp$ then $r \xleftarrow{\$} \mathbb{Z}_p$; $Add(m, PK_A; PK_B, \star, K, r)$ to L_1.
return r

Fig. 2. Random Oracle Simulators $\mathsf{H_0Sim}$ and $\mathsf{H_1Sim}$

$$Pr[S_1] = Pr[S_0]$$

Game $\mathbf{G_2}$: In this game, we replace the signcryption oracle by the signcryption oracle simulator SCSim as described in Figure 3. This simulator does not require the sender's private key $priv_S$ to generate a signcryptext. Since s, r are uniformly chosen at random in $\mathbb{Z}_p$ and W is computed as $W = [s]PK_S + [r]G$, W is therefore uniformly distributed in $\mathbb{G}$. As a result, as long as $\perp_{SC}$ does not occur, we have that Game 1 and Game 2 are equivalent. Note that the size of L_i is bounded by $(q_i + q_{SC} + q_{USC})$ for $i \in \{0,1\}$. Thus, the probability that $\perp_{SC}$ happens is bounded by $(q_1 + q_{SC} + q_{USC})/2^{l_p(k)}$ and there are at most q_{SC} executions. Hence, we have:

$$|Pr[S_2] - Pr[S_1]| \leq q_{SC}\left(\tfrac{q_1+q_{SC}+q_{USC}}{2^{l_p(k)}}\right)$$

$\mathsf{SCSim}(PK_A, (PK_B, m))$:
$s; r \xleftarrow{\$} \mathbb{Z}_p$, $W = [s]PK_S + [r]G$;
$\tau \leftarrow L_0Rule(PK_A, PK_B, null, W)$
If $\tau = \perp$ then $\tau \xleftarrow{\$} \{0,1\}^l$; Add $(PK_A, PK_B, W, \star, \tau)$ to L_0
$c \leftarrow Enc_\tau(m)$, $r' = L_1Rule(m, PK_A, PK_B, null, W)$
If $r' \neq \perp$ then return $\perp$ and halt all operations (event $\perp_{SC}$)
else Add $(m, PK_A, PK_B, W, \star, r)$ to L_1; $C \leftarrow (r, s, c)$
return C

Fig. 3. Signcryption Oracle Simulator SCSim

$\mathsf{USCSim}(PK_B, (C, PK_A))$:
Parse C as (r, s, c)
$W = [s]PK_A + [r]G$; $\tau \leftarrow L_0Rule(PK_A, PK_B, null, W)$
If $\tau = \perp$ then $\tau \xleftarrow{\$} \{0,1\}^l$; Add $(PK_A, PK_B, W, \star, \tau)$ to L_0
$m = Dec_\tau(c)$; $r' = L_1Rule(m, PK_A, PK_B, null, W)$
If $r' = \perp$ then $r' \xleftarrow{\$} \mathbb{Z}_p$; Add $(m, PK_A, PK_B, W, \star, r')$ to L_1 if $r \neq r'$ then return $\perp$ else return m

Fig. 4. Unsigncryption Oracle Simulator USCSim

Game $\mathbf{G_3}$: This game replaces the unsigncryption oracle by the simulator USCSim described in Figure 4, in order not to use the receiver's private key $priv_R$. We observe that Game 3 is identical to Game 2 except when the hash oracles are queried at $K^* = [priv_R]^{-1}([s]PK_S + [r]G)$. We consider this situation in three cases:

– H_0 is queried on (PK_S, PK_R, K^*) or H_1 is queried on (m, PK_S, PK_R, K^*) by the adversary $\mathcal{A}$. This means that $\mathcal{A}$ can recover K^*. As a result, this leads to an algorithm $\mathcal{B}_1$ that can solve the GDH problem, because the adversary can verify the fact that $\mathcal{O}(PK_R, W, K^*) = 1$.

– The signcryption oracle could attempt to make such queries. However, this requires that the value of W must be equal to W^*. Since r, s are uniformly chosen at random in $\mathbb{Z}_p$, the probability that this event occurs, is bounded by the probability $q_{SC}/2^{l_p(k)}$.

– The unsigncryption oracle could attempt to make such queries. In such situation, the adversary must have made a query to $\mathcal{O}_{USC}$ on (c, r, s) such that: $[s]PK_S + [r]G = [s^*]PK_S + [r^*]G$ (1). If $(s, r) = (s^*, r^*)$ then we must have $c \neq c^*$, because $\mathcal{A}$ is not allowed to query exactly to $\mathcal{O}_{USC}$ on the signcryptext obtained from the signcryption oracle. We must also have $\tau = \tau^*$. Since the symmetric encryption scheme is one-to-one, we obtain that $(m = Dec_\tau(c)) \neq (m_b = Dec_{\tau^*}(c^*))$. As a result, this equation must hold $H_1(PK_S||PK_R||K^*||m) = H_1(PK_S||PK_R||K^*||m_b)$. However, as H_1 is modeled as a random oracle, the equation is true only with probability of $1/2^{l_p(k)}$. We then change the unsigncryption oracle so that it answers $\perp$ when queried on (c, r^*, s^*). The probability that it outputs incorrectly is bounded by $q_{SC}/2^{l_p(k)}$. On the other hand, if $(s, r) \neq (s^*, r^*)$, we show that the GDH problem can be solved. In fact, from (1), we obtain that $[(s - s^*)]PK_S = [(r^* - r)]G$. We can deduce that $priv_S{}^{-1} = (r^* - r)/(s - s^*)$. Hence, one can compute $[ab]G = [priv_S^{-1}]PK_R = [(r^* - r)/(s - s^*)]PK_R$. At this stage, $\mathcal{A}$ can verify the accuracy of $[ab]G$ by using the DDH oracle $\mathcal{O}$. Consequently, we have:

$$|Pr[S_3] - Pr[S_2]| \leq \frac{q_S + q_{USC}}{2^{l_p(k)}} + Adv^{GDH}_{\mathcal{B}_1}$$

In G_3, τ^* is not used anywhere except when computing the challenge ciphertext c^*. Hence, if $\mathcal{A}$ outputs $b' = b$, then there exists an algorithm $\mathcal{B}_2$ that can break the OT-IND property of the symmetric encryption scheme. Thus, $Pr[S_3] = \frac{1}{2} + \frac{1}{2}Adv^{OT-IND}_{\mathcal{B}_2}(k)$. Summarizing all the obtained bounds together, we have: $Adv^{SC-IND-CCA2}_{\mathcal{A}}(k) = 2|Pr[S_0] - \frac{1}{2}| \leq \frac{2q_{SC}(q_1+q_{SC}+q_{USC})}{2^{l_p(k)}} + \frac{2(q_{SC}+q_{USC})}{2^{l_p(k)}} + 2Adv^{GDH}_{\mathcal{B}_1}(k) + Adv^{OT-IND}_{\mathcal{B}_2}(k)$. ∎

C. Unforgeability of our scheme

Theorem 10. *In the random oracle model, given a PPT adversary $\mathcal{A}$ against the SC-UF-CMA property of the proposed signcryption scheme, there exists a PPT algorithm $\mathcal{B}$ against the GDL problem such that: $Adv^{SC-UF-CMA}_{\mathcal{A}}(k) \leq \sqrt{q_{\mathcal{R}} \cdot Adv^{GDL}_{\mathcal{B}}(k)} + \frac{q_{SC}(q_1+q_{SC})+q_{\mathcal{R}}+1}{2^{l_p(k)}}$.*

We prove the theorem using two lemmas. First, we show that if there exists an attacker $\mathcal{A}$ against the SC-UF-CMA property, we can construct an efficient algorithm $\mathcal{B}'$ that solves the GDL' problem which is defined below. Then, we prove that any efficient algorithm $\mathcal{B}'$ can be transformed to an efficient algorithm $\mathcal{B}$ that solves the GDL problem, thus contradicting with the hardness assumption of GDL in section II-B1.

Definition 11 (GDL' problem)**.** *Given $(G, n, p, [a]P)$, where $(G, n, p) \xleftarrow{\$} \mathsf{Setup}(k)$ and $a \xleftarrow{\$} \mathbb{Z}_p$, we define an oracle $\mathcal{R}$ as follows: for i=1..$q_{\mathcal{R}}$, on input $(PK_i, K_i) \in \mathbb{G}^2$, return $r_i \xleftarrow{\$} \mathbb{Z}_p$, where $q_{\mathcal{R}}$ is the maximum number of queries made to $\mathcal{R}$. The GDL' problem is to compute s^* and $i^* \in \{1..q_{\mathcal{R}}\}$ such that: $K_{i^*} = [as^* + r_{i^*}]PK_{i^*}$.*

We first reduce the hardness of SC-UF-CMA property to the hardness of the GDL' problem as follows:

Lemma 12. *If there exists a PPT adversary $\mathcal{A}$ against the SC-UF-CMA property, then there exists a PPT adversary $\mathcal{B}'$ against the GDL' problem, such that: $Adv^{SC-UF-CMA}_{\mathcal{A}}(k) \leq Adv^{GDL'}_{\mathcal{B}'}(k) + \frac{q_{SC}(q_1+q_{SC})+1}{2^{l_p(k)}}$.*

Proof: We will prove the lemma via a sequence of game [35]. At the end of each game, $\mathcal{A}$ outputs a tuple consisting of $(priv^*_R, PK^*_R, C^*)$. Let Verify be the algorithm that verifies the two conditions listed in Definition 8. We denote S_i is the event in game G_i that Verify outputs 1.

Game $\mathbf{G_0}$: This is the original attack game $EXPT^{SC-UF-CMA}_{\mathcal{A}}$ in Definition 8. Hence,

$$Pr[S_0] = Adv^{SC-UF-CMA}_{\mathcal{A}}(k)$$

Game $\mathbf{G_1}$: This game replaces the random oracles H_0 and H_1 by the simulated oracles $\mathsf{H_0Sim}$ and $\mathsf{H_1Sim}$. $\mathsf{H_0Sim}$ remains unaltered as described in Figure 2, while $\mathsf{H_1Sim}$ is modified as described in Figure 5. The lists L_0 and L_1 are still employed to store historical queries on simulated oracles. The rules for determining membership of these lists remain unchanged. As we shall see, $\mathsf{H_1Sim}$ makes call to the oracle $\mathcal{R}$ defined in the GDL' problem. Note that $\mathcal{R}$ behaves differently from a random oracle, because it always returns random values even for repeated queries. Besides, we introduce the list $L_{\mathcal{R}}$ that contains the values of type $(PK_B, K, r, j) \in \mathbb{G}^2 \times \mathbb{Z}_p \times \mathbb{Z}$. The above simulation for the random oracle H_0 and H_1 is perfect. Hence, we have

$$Pr[S_1] = Pr[S_0]$$

Game $\mathbf{G_2}$: This game replaces the signcryption oracle by the simulated oracle simulator SCSim described in Figure 3. This simulator does not require the sender's private key $priv_S$ in the signcryption stage.

$\mathsf{H_1Sim}(m, PK_A, PK_B, K)$:
$r \leftarrow L_1 Rule(m, PK_A, PK_B, K, null)$
If $r = \perp$ then $j \leftarrow j + 1$; $r \xleftarrow{\$} \mathcal{R}(PK_B, K)$; Add (PK_B, K, r, j) to $L_{\mathcal{R}}$; $Add(m, PK_A; PK_B, \star, K, r)$ to L_1.
return r

Fig. 5. Random Oracle Simulators $\mathsf{H_1Sim}$ in game G_1

Since (s, r, W) are independent and uniformly distributed over $\mathbb{Z}_p^2 \times \mathbb{G}$, the views of attacker in Game G_1 and Game G_2 are equivalent, as long as the event $\perp_{SC}$ does not happen. The size of L_i is bounded by $(q_i + q_{SC})$ for $i \in 0, 1$. Thus, the probability that $\perp_{SC}$ happens is bounded by $(q_1 + q_{SC})/2^{l_p(k)}$. There are maximum of q_{SC} queries on the signcryption oracle. Hence, we have

$$|Pr[S_2] - Pr[S_1]| \leq q_{SC}\left(\frac{q_1+q_{SC}}{2^{l_p(k)}}\right)$$

Now, we consider the event AskKey that $\mathsf{H_1Sim}$ has been queried on (m^*, PK_S, PK_R, K^*). Note that if AskKey does not occur, then the value r returned by $\mathsf{H_1Sim}$ is uniformly generated from $\mathbb{Z}_p$. If C^* is a valid signcryptext then $H_1(m^*, PK_S, PK_R, K^*)$ must not have been defined by the signcryption oracle. Thus, the probability that $r = H_1(m^*, PK_S, PK_R, K^*)$ is at most $1/2^{l_p(k)}$. As a result,

we obtain that $Pr[S_2|\neg\mathsf{AskKey}] \leq 1/2^{l_p(k)}$ and consequently $Pr[S_2] \leq Pr[\mathsf{AskKey}] + 1/2^{l_p(k)}$.

On the other hand, we show that if AskKey occurs, then there exists an algorithm $\mathcal{B}'$ against the GDL' problem. Indeed, $\mathcal{B}'$ is given inputs (G, n, p, PK_S) and runs $\mathcal{A}$ on this input. If AskKey occurs, then $\mathcal{A}$ must return (PK_R, r^*, s^*, c^*) such that $\mathsf{H_1Sim}$ is queried on (m^*, PK_S, PK_R, K^*). Since (m^*, PK_R) has never been queried to SCSim, $\mathcal{R}$ must be queried on (PK_R, K^*) by $\mathsf{H_1Sim}$ and return r^*. Thus, there will exist an entry $(PK_R, K^*, r^*, j) \in L_{\mathcal{R}}$, where $1 \leq j \leq q_1$. As a result, (s^*, j) is a valid solution for the GDL' problem. Therefore, we have $Pr[\mathsf{AskKey}] \leq Adv_{\mathcal{B}'}^{GDL'}(k)$. In conclusion, we achieve the following reduction:

$$Adv_{\mathcal{A}}^{SC-UF-CMA}(k) \leq Adv_{\mathcal{B}'}^{GDL'}(k) + \frac{q_{SC}(q_1+q_{SC})+1}{2^{l_p(k)}}$$

■

In the following, we will apply the general forking lemma defined by Bellare et al. in [8] to reduce GDL' to GDL. This approach is also used by Zhang et al. [7] in their proof. We recall the general forking lemma as follows:

Lemma 13 (General forking lemma [8])**.** *Fixing an integer $q_{\mathcal{R}} \geq 1$ and a set Z of size $h = 2^{l_p(k)} \geq 2$. Let $\mathcal{V}$ be a randomized algorithm that on input $(cp, r_1, r_2, ..., r_{q_R})$ returns a pair (J, σ) consisting of an integer $0 \leq J \leq q_{\mathcal{R}}$ and a side output σ. Let IG be a randomized algorithm that we call input generator. The accepting probability of $\mathcal{V}$, denoted as acc, is defined as the probability that $J \geq 1$ in the experiment: $cp \xleftarrow{\$} IG; r_1, r_2, ..., r_{q_R} \xleftarrow{\$} Z; (J, \sigma) \xleftarrow{\$} V(cp, r_1, r_2, ..., r_{q_R})$. The forking algorithm associated to $\mathcal{V}$ is defined as follows: $F_{\mathcal{V}}(cp)$:*

Pick coins ρ for $\mathcal{V}$ at random $r_1, ..., r_{q_R} \xleftarrow{\$} Z$; $(I, \sigma) \leftarrow \mathcal{V}(cp, r_1, ..., r_{q_{\mathcal{R}}}; \rho)$; If $I = 0$ return $(0, \varepsilon, \varepsilon)$; $r'_1, ..., r'_{q_R} \xleftarrow{\$} Z$; $(I', \sigma') \leftarrow \mathcal{V}(cp, r_1, ..., r_{I-1}, r_I, ..., r_{q_{\mathcal{R}}}; \rho)$
If $I = I'$ and $r_I \neq r'_I$ return $(1, \sigma, \sigma')$ else return $(0, \varepsilon, \varepsilon)$
Let

$$frk = Pr[b = 1 : cp \xleftarrow{\$} IG; (b, \sigma, \sigma') \xleftarrow{\$} F_{\mathcal{V}}(cp)]$$

Then $frk \geq acc \cdot (\frac{acc}{q_{\mathcal{R}}} - \frac{1}{h})$ and alternatively

$$acc \leq \frac{q_{\mathcal{R}}}{h} + \sqrt{q_{\mathcal{R}} \cdot frk}$$

Lemma 14. *If there exists a PPT adversary $\mathcal{B}'$ against the GDL' problem, then there exists a PPT adversary $\mathcal{B}$ against the GDL problem such that: $Adv_{\mathcal{B}'}^{GDL'}(k) \leq \frac{q_{\mathcal{R}}}{h} + \sqrt{q_{\mathcal{R}} \cdot Adv_{\mathcal{B}}^{GDL}(k)}$*

Proof: We will use the general forking lemma in this proof. As defined in the proof of Lemma 12, $\mathcal{B}'$ is the algorithm that can solve the GDL' problem. It takes as input (G, n, p, PK_S) where $a = priv_S^{-1}$, and returns (j^*, s^*, r^*) or $\perp$. We denote an algorithm $\mathcal{V}$ that runs $\mathcal{B}'$ as a subroutine. It takes as input $(G, n, p, PK_S, r_1, ..., r_{q_{\mathcal{R}}})$. It outputs values of type (j, σ) or $(0, \varepsilon, \varepsilon)$, where σ is a tuple of the form $(s, r) \in \mathbb{Z}_p^2$. The forking algorithm $F_{\mathcal{V}}$ is built as in Lemma 13. We define an algorithm $\mathcal{B}$ that runs $F_{\mathcal{V}}$ as a subroutine. If $F_{\mathcal{V}}$ returns $(1, \sigma, \sigma')$, such that: $\sigma = (s^*, r^*)$ and $\sigma' = (s^{*\prime}, r^{*\prime})$, we have $K^* = K^{*\prime}$ and $PK_{i^*} = PK_{i^{*\prime}}$ (because $j^* = j^{*\prime}$). As a result, the following equation holds: $[priv_S^{-1} \cdot s^* + r_{j^*}]PK_{i^*} = [priv_S^{-1} \cdot s^{*\prime} + r_{j^{*\prime}}]PK_{i^{*\prime}}$. Since $r_{j^*} \neq r_{j^{*\prime}}$ as defined in the forking algorithm $F_{\mathcal{V}}$, we can extract the sender's private key as follows: $priv_S = (s^* - s^{*\prime})/(r_{j^{*\prime}} - r_{j^*})$. Then $\mathcal{B}$ outputs $priv_S$ as a solution for an instance of the GDL problem. As we can see, $\mathcal{V}$ outputs essentially what $\mathcal{B}'$ outputs. It is obvious that the accepting probability acc is equal to the success probability of $\mathcal{B}'$, $Adv_{\mathcal{B}'}^{GDL'}(k)$. Similarly, $\mathcal{B}$ outputs identically as $F_{\mathcal{V}}$, so that $Adv_{\mathcal{B}}^{GDL}(k) = frk$. Hence, by the general forking lemma, we have:

$$Adv_{\mathcal{B}'}^{GDL'}(k) \leq \frac{q_{\mathcal{R}}}{h} + \sqrt{q_{\mathcal{R}} \cdot Adv_{\mathcal{B}}^{GDL}(k)}$$

■

V. Provided security features and extension

We have formally proved in section IV that our scheme SCKWC is confidentially secure in the outsider model and unforgeably secure in the insider model. In [3], the authors suggest that a signcryption scheme should also support the "public verifiability" and "non-repudiation" features. We claim that SCKWC provides these properties.

Public verifiability: To prove to a trusted third party that the sender S actually signed the plaintext m, R can forward the following tuple $(PK_S, PK_R, m, K, r, s, c)$. The third party can verify the signcryptext by executing the steps belows:

- Compute $\tau = H_0(PK_S||PK_R||K)$
- Verify if $m = Dec_\tau(c)$.
- Verify if $r = H_1(PK_S||PK_R||K||m)$

The knowledge on K does not leak any secrecy on the private key of either S or R, as long as the DLP assumption remains unbreakable for any resource-bounded attackers.

Non-repudiation: The non-repudiation is a direct result of the unforgeability feature. The sender usually can not deny the authority of the signcryptext when executing the above *public verifiability* process, if the ciphertext is really issued by him. However, if the aforementioned process passes successfully, then duplicating valid signcryptext is possible, which is inconsistent to the unforgeability feature.

It is possible to add the property of insider confidentiality to the previous proposed scheme with the cost of an extra point multiplication. This property was also considered in [6], [9], [7] and called "forward security" in several existing works [16], [13], [15]. Indeed, instead of returning (r, s, c), Signcrypt returns (Q, s, c), where $Q = [r]G$. Similarly, Unsigncrypt verifies the validity of Q instead of r, as follows: $Q \stackrel{?}{=} [H_1(PK_S||PK_R||K||m)]G$. As we can see, it is now computationally infeasible for a bounded resource adversary to recover messages of previous sessions even under exposure of the private key of the sender due to the DLP assumption. We name the resulting scheme as SCKWC+.

VI. Performance evaluation

This section first quantifies the performance of our proposed schemes and then estimates their energy consumption versus other related schemes on an emulated sensor platform.

A. Performance comparison

Table I illustrates the efficiency and supported security features of our schemes and multiple signcryption proposals in related work. The table shows if the scheme supports certificateless property. Then, the efficiency of each scheme is evaluated with respect to: communication and computational costs. The communication costs are evaluated as the packet length of signcryptext in bits. While, the computational costs are evaluated in terms of the number of expensive operations needed for the signcryption and unsigncryption processes. Finally, the table summarizes the supported security properties for each scheme.

As shown in Table I, our proposed schemes not only support desirable security features, but also offer the best

Scheme	CL	Efficiency: Communication cost	Computational cost: Signcryption PM	PA	I	e	EXP	Unsigncryption PM	PA	I	e	EXP	Supported features: UF	OCF	NR	PV	ICF	StS
Zheng [3]	❍	$2\|p\|+\|m\|$	0	0	1	0	1	0	0	0	0	2	●	●	●	●	❍	n/a
SCDSA+ [12]	❍	$2\|p\|+\|m\|$	0	0	2	0	2	0	0	1	0	3	●	●	●	●	❍	DSA
Bao et al. [14]	❍	$2\|p\|+\|m\|$	0	0	1	0	2	0	0	0	0	3	❍	●	●	●	❍	n/a
Yum et al. [5]	❍	$2\|p\|+\|m\|$	0	0	0	0	1	0	0	0	0	3	❍	●	●	●	❍	KCDSA
Selvi et al. [24]	●	$2\|p\|+\|m\|$	0	0	0	0	5	0	0	0	0	7	●	●	●	❍	●	n/a
S-ECSC [4]	❍	$2\|p\|+\|m\|$	1	0	0	0	0	3	1	0	0	0	●	●	●	●	❍	n/a
ECGSC [22]	❍	$\|G\|+\|p\|+\|m\|$	2	0	1	0	0	3	1	1	0	0	●	●	●	●	●	ECDSA
NCLSC [28]	●	$3\|G\|+\|m\|$	3	1	0	0	0	2	2	0	2	0	●	●	●	●	●	n/a
Tso et al. [13]	❍	$\|G\|+\|p\|+\|m\|$	3	0	1	0	0	4	1	3	0	0	●	●	●	●	●	ECDSA
Toorani et al. [16]	❍	$\|G\|+\|p\|+\|m\|$	2	0	0	0	0	4	2	0	0	0	●	●	●	●	●	n/a
Dutta et al. [15]	❍	$\|G\|+\|p\|+\|m\|$	3	0	1	0	0	5	2	0	0	0	●	●	●	●	●	n/a
SCKWC	●	$2\|p\|+\|m\|$	1	0	0	0	0	3	1	0	0	0	●	●	●	●	❍	KCDSA
SCKWC+	●	$\|G\|+\|p\|+\|m\|$	2	0	0	0	0	3	1	0	0	0	●	●	●	●	●	KCDSA

TABLE I. PERFORMANCE COMPARISON BETWEEN OUR SCHEME AND RELATED WORK

Meaning of abbreviations: CL: Certificateless or Public key Verification without a trusted third party, PM: Point multiplication, PA: Point addition, EXP: Modular exponentiation, I: Modular inversion, e: Pairing operation, UF: Unforgeability, OCF: Outsider Confidentiality, NF: Non-repudiation, PV: Public verifiability, ICF: Insider Confidentiality or Forward secrecy, StS: Standard signature. We define simple symbols to evaluate the security services: ●- supported, ❍- not supported. The n/a notation means "not applicable". $|Y|$ denotes the length of Y in bits.

performance in terms of computational cost. Indeed, SCKWC requires only 1 point multiplications (PM) for signcryption, 3 PMs and one point addition for unsigncryption. SCKWC+ requires one more point multiplication in the signcryption process. When compared to the other elliptic curve based schemes ([15], [16], [13], [28], [22]), SCKWC+ needs the least number of costly operations and also generates the shortest signcrytext in bits.

B. Estimation of energy consumption on emulated sensor platform

In the next subsection, we provide details on the implementation of our performance assessment. Thereafter, we report the performance and energy consumption results of our scheme compared with related work.

1) Experimental tools and platforms: We have implemented our assessment in C for the operating system Contiki 2.7 [33]. Based on the Relic library [37] version 0.3.5, we evaluate some cryptographic operations on the four elliptic curves secg_p160, nist_p192, nist_p224 and nist_p256. Their domain parameters have been recommended by SECG [38] and NIST [21]. In addition, we opted for the emulated sensor node Wismote to evaluate the required operations on Cooja [31] - a Java-based simulator designed for the Contiki operating system. Wismote [39] is a low power wireless sensor module featured with 16 MHz MSP430x micro-controller, 16 kB of RAM, 128 kB of ROM and an IEEE 802.15.4 radio interface. This platform supports 20 bit addressing and sufficient RAM and ROM capacities. Such features are necessary for using a cryptographic library along with an application on top of it.

2) Performance: In order to assess the energy consumption, we employ a software-based online energy estimation mechanism described in [34]. In their model, the total energy consumption can be evaluated by the following formula: $E = U * (I_m t_m + I_l t_l + I_t t_t + I_r t_r + \sum I_{c_i} t_{c_i})$, where U is the supply voltage, I_i and t_i $(i = m, l, t, r)$ are the current draw and the time duration of the microprocessor in active mode, low power mode, transmit mode and receive mode respectively. I_{c_i} and t_{c_i} are the current draw and the time duration of the microprocessor for handling other components, such as sensors and LEDs.

In our scenario, we consider only the first four factors. The value of U is typically 3V, as with two new AA batteries. Furthermore, the current draw of the sensor node in each mode is extracted from its data sheet. As an example, the Wismote platform consumes I=2.2mA when in active mode. The time t_i that the component is in mode i, is measured by Powertrace. The latter is a pre-loaded tool in the Contiki OS, which provides an accuracy up to 94% of the energy consumption of a device [32].

Table II shows the execution time and energy cost of ECC operations over the Wismote platform. We consider only the most expensive operations: point multiplication(PM), point addition(PA), modular inversion and pairing operation. Each operation is evaluated in the four mentioned elliptic curves in increasing order of security level. Pairing-based calculation is, as expected, the most expensive operation. Point multiplication is also an expensive task. That is, even for the smallest security level of 80 bits, it requires up to 2.5s to compute and consumes 16.25mJ. In addition, we observe that for an elliptic curve with length of 256 bits of field order, the energy cost for point multiplications and pairing operations becomes huge, since for a single execution, they consume more than 124mJ and 239mJ, respectively. Besides, they are also time-consuming (18.91s for a PM and 36.16s for a pairing).

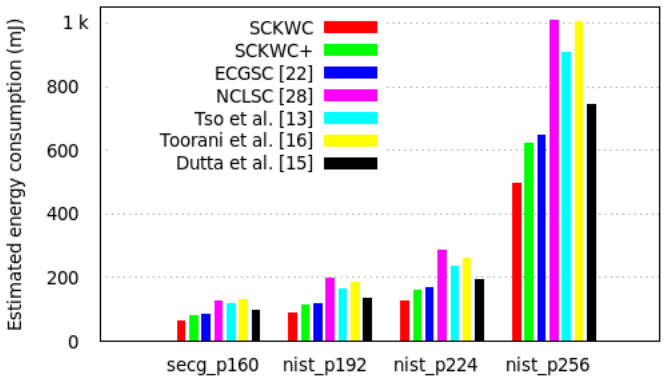

Fig. 6. Total estimated energy consumption of our schemes and related work

Gathering the measurement results in Table I and II, we estimate the total energy consumption of our proposed signcryption schemes and five other ECC-based signcryption schemes. As depicted in Figure 6, our proposals SCKWC and SCKWC+ are the most efficient schemes. The ECGSC [22] scheme has a slightly higher computational cost in comparison with ours. However, it requires certificates to validate the public keys. This constraint could be very costly for a sensor node, since the verification of certificates is usually

Parameters	Strength	Size	PM	PA	Inversion	Pairing
secg_p160	80	160	2460ms/16.25mJ	7ms/0.03mJ	298ms/1.90mJ	3533ms/23.32mJ
nist_p192	96	192	3463ms/22.53mJ	8ms/0.04mJ	403ms/2.67mJ	6586ms/43.47mJ
nist_p224	112	224	4782ms/32.05mJ	10ms/0.07mJ	577ms/3.81mJ	9573ms/63.19mJ
nist_p256	128	256	18.91s/124.07mJ	31ms/0.21mJ	1870ms/12.36mJ	36,16s/238.13mJ

TABLE II. ENERGY CONSUMPTION AND TIME EXECUTION OF ATOMIC OPERATIONS ON WISEMOTE

complicated and consuming in energy. Indeed, SCKWC+ saves more than 17%, 31%, 38% and 41% of the overall energy consumption in comparison with the schemes of Dutta et al. [15], Tso et al. [13], Toorani et al. [16] and NCLSC [28], respectively. SCKWC is even more efficient than SCKWC+ and therefore can be applied on resource-constrained devices.

VII. RELATED WORK

In this paper, we are mainly interested in the signcryption schemes based on the Diffie-Hellman problem. As surveyed in [17], there exist several schemes based on different security assumptions, such as: Bilinear Maps [18] and RSA problem [19]. Most of the signcryption schemes are derived from popular signature schemes. Zheng's scheme [3] is based on Elgamal encryption and signature [20], which is computationally efficient, but requires complex interactive zero-knowledge proof to validate the non-repudiation and does not provide insider confidentiality. Bao et al. [14] modify Zheng's proposal to provide the public verifiability property without the need for the recipient's private key. However, the previous scheme is not semantically secure, as written by Shin et al. [12]. They claim their new signcryption proposal based on DSA (Digital Signature Algorithm) [21], namely SCDSA+, to be confidentially and unforgeably secure, without giving a formal proof. There exist also several schemes issued from the standardized signature algorithm ECDSA [22], [13]. Both schemes provide desirable security properties as depicted in Table I but still result in poorer performance than our schemes. Certificateless signcryption schemes remove the use of certificates. However, they usually require costly pairing operations for public key validation [26], [27]. Some similar proposals are successful to remove pairing operations in their construction [25], [24]. However, they still require 10 and 12 modular exponentiations.

Two signcryption variants of KCDSA are first proposed by Yum et al. [5]. However, their security has not been formally proved by the authors. Besides, the first variant is confidentially insecure in the insider model. The second one is not semantically secure due to the disclosure on the hash of the message, in addition to being more expensive in terms of performance comparing to our first proposal SCKWC (one extra exponentiation). Several works on identity-based signcryption scheme based on KCDSA exist, such as [10], [11]. Though, these schemes require 3 costly pairing operations, which is not practical for constrained nodes in the IoT.

VIII. CONCLUSION

This paper proposed two lightweight signcryption schemes derived from the standardized signature KCDSA that do not require the use of certificates. The first proposal SCKWC has been formally proved to be outsider confidentially and insider unforgeablely secure against chosen ciphertext/message attacks in the random oracle model. The second variant is secure in the insider model but requires one more point multiplication. Furthermore, our schemes offer efficiency both in terms of communication and energy consumption costs. The efficiency of the proposed schemes has been validated by an experimental evaluation on an emulated sensor platform. As future work, we plan to integrate the proposed signcryption schemes into a security framework designed for the IoT.

REFERENCES

[1] M. Groves, *Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)*, RFC 6507.

[2] C. H. Lim et al., *A Study on the Proposed Korean Digital Signature Algorithm*, ASIACRYPT'98, September 2002.

[3] Y. Zheng, *Digital Signcryption or How to Achieve Cost (Signature & Encryption) << Cost(Signature) + Cost(Encryption)*, Crypto'97, LNCS vol. 1294, pp. 165-179, 1997.

[4] X. Zhou et al., *Short Signcryption Scheme for the Internet of Things*, Informatica 35:521530, 2011.

[5] D. H. Yum et al., *New Signcryption Schemes Based on KCDSA*, in ICICS 2001, LNCS 2288, pp. 305317, 2002.

[6] J. Baek et al., *Formal Proofs for the Security of Signcryption*, PKC'02, 2002.

[7] Wei Zhang, *Improvements and Generalisations of Signcryption Schemes*, Doctoral Thesis, Royal Holloway, University of London, 2013.

[8] M. Bellare et al., *Multi-signatures in the plain public-key model and a general forking lemma*, in CCS 06, pp. 390-399, 2006.

[9] J. H. An et al., *On the security of joint signature and encryption*, Eurocrypt'02, LNCS 2332, pp. 83-107, 2002.

[10] F. Li et al., *An Improved Identity-Based KCDSA Signcryption Scheme*, ISDPE, 2007.

[11] J.H. Ryu et al., *Identity based KCDSA signcryption*, ICACT 2006, Vol. 2, pp. 13691374, 2006.

[12] J. Shin et al., *New DSA-Verifiable Signcryption Schemes*, ICISC, 2002.

[13] R. Tso et al., *ECDSA-Verifiable Signcryption Scheme with Signature Verification on the Signcrypted Message*, Information Security and Cryptology, LNCS 4990, pp. 11-24, 2008.

[14] F. Bao et al., *A Signcryption scheme with signature directly verifiable by public key*, PKC'98, 1998.

[15] M. Dutta et al., *An Efficient Signcryption Scheme based on ECC with Forward Secrecy and Encrypted Message Authentication*, IACC, 2013.

[16] M. Toorani et al., *An elliptic curve-based signcryption scheme with forward secrecy*, Journal of Applied Sciences, 9(6):1025-1035, 2009.

[17] A. W. Dent and Y. Zheng, editors. *Practical Signcryption*, Springer-Verlag, 2010.

[18] P. S. L. M. Barreto et al., *Efficient and provably-secure identity-based signatures and signcryption from bilinear maps*, Asiacrypt 2005, LNCS vol. 3788, pp. 515-532, 2005.

[19] A. W. Dent et al., *Signcryption Schemes Based on the RSA Problem*, In, A. W. Dent and Y. Zheng, editors, Practical Signcryption, pp. 99-120, 2010.

[20] T. ElGamal, *A public key cryptosystem and a signature scheme based on discrete logarithms*, Crypto 84, vol. 196, LNCS, pp. 1018, 1984.

[21] NIST (National Institute for Standard and Technology), *Digital Signature Standard (DSS)*, FIPS PUB 186, 1994.

[22] Y. Han et al., *ECGSC: Elliptic Curve Based Generalized Signcryption*, 3^{rd} ICUIC, LNCS, vol. 4159, pp 956-965, 2006.

[23] M. Groves, *MIKEY-SAKKE: Sakai-Kasahara Key Encryption in Multimedia Internet KEYing (MIKEY)*, RFC 6509, February 2012.

[24] S. S. D. Selvi et al., *Cryptanalysis of Certificateless Signcryption Schemes and an Efficient Construction without Pairing*, Inscrypt, LNCS 6151, pp. 75-92, 2010.

[25] W. Xie et al., *Certificateless signcryption without pairing*, in Cryptology ePrint Archive: Report 2010/187,http://eprint.iacr.org/2010/187.pdf.

[26] M. Barbosa et al., *Certificateless Signcryption*, ASIACCS, 2008.

[27] C. Wu et al., *A new efficient Certificateless Signcryption scheme*, ISISE, 2008.

[28] G. Yu et al., *Efficient Certificateless Signcryption Scheme from Weil Pairing*, Journal of Networks, vol. 6, no. 9, 2011.

[29] S. S. Al-Riyami et al., *Certificateless Public-Key Cryptography*, ASIACRYPT 2003, LNCS 2894, pp. 452-473, 2003.

[30] A. Shamir, *Identity-based cryptosystems and signature shemes*, in Proc. Crypto84, pp.47-54, 1984.

[31] J. Eriksson et al., *COOJA/MSPSim: interoperability testing for wireless sensor networks*, Proc. of the 2nd International Conference on Simulation Tools and Techniques, 2009.

[32] A. Dunkels et al., *Powertrace: Network-level power profiling for low-power wireless networks*, Technical Report T2011:05, Swedish Institute of Computer Science, 2011.

[33] A. Dunkels et al., *Contiki - a lightweight and flexible operating system for tiny networked sensors*, In Workshop on Embedded Networked Sensors, 2004.

[34] A. Dunkels et al., *Software-based on-line energy estimation for sensor nodes*, Emnets IV, 2007.

[35] V. Shoup, *Sequences of games: a tool for taming complexity in security proofs*, manuscript. Revised Jan. 18, 2006, http://www.shoup.net/papers/games.pdf.

[36] M. Bellare et al., *Random Oracles Are Pratical: a Paradigm for Designing Efficient Protocols*, In Proc. of the first CCCS, pages 62-63, 1993.

[37] Relic toolkit, an efficient library for cryptography, https://code.google.com/p/relic-toolkit/.

[38] SECG. Sec 2: Recommended elliptic curve domain parameters version 2.0.

[39] Arago Systems, WiSMote, http://www.aragosystems.com/images/stories/WiSMote/Doc/wismote_en.pdf, last accessed: 01.02.2015.

[40] NSA, *The Case for Elliptic Curve Cryptography*, http://www.nsa.gov/business/programs/elliptic_curve.shtml, accessed Dec. 2014.