1-1-2008

Identity-based on-line/off-line signcryption

Dongdong Sun
University of Wollongong

Xinyi Huang
University of Wollongong, xh068@uow.edu.au

Yi Mu
University of Wollongong, ymu@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Identity-based on-line/off-line signcryption

Abstract

We present an identity-based on-line/off-line signcryption scheme, where most of computations are carried out when the message is not available(i.e., off-line stage) and the on-line part of our scheme does not require any exponent computations and therefore is very efficient. It combines the functionalities of signature and encryption and is provably secure in the random oracle model. We also show that our scheme is indistinguishable against adaptive chosen-ciphertext attacks (IND-IDSC-CCA2) and is existentially unforgeable against adaptive chosen-message attacks (EF-IDSC-ACMA).

Keywords

Identity, based, line, off, line, signcryption

Disciplines

Business | Social and Behavioral Sciences

Publication Details

Sun, D., Huang, X., Mu, Y. & Susilo, W. (2008). Identity-based on-line/off-line signcryption. In J. Cao, M. Li, C. Weng, Y. Xiang, X. Wang, H. Tang, F. Hong, H. Liu & Y. Wang (Eds.), IFIP International Conference on Network and Parallel Computing (pp. 34-41). USA: IEEE.

Identity-Based On-line/Off-line Signcryption

Dongdong Sun, Xinyi Huang, Yi Mu and Willy Susilo
School of Computer Science and Software Engineering
University of Wollognong
Wollongong, NSW 2522, Australia
{dds03, xh068, ymu, wsusilo}@uow.edu.au

Abstract

We present an identity-based on-line/off-line signcryption scheme, where most of computations are carried out when the message is not available(i.e., off-line stage) and the on-line part of our scheme does not require any exponent computations and therefore is very efficient. It combines the functionalities of signature and encryption and is provably secure in the random oracle model. We also show that our scheme is indistinguishable against adaptive chosen-ciphertext attacks (IND-IDSC-CCA2) and is existentially unforgeable against adaptive chosen-message attacks (EF-IDSC-ACMA).

1. Introduction

In public key systems, the authenticity of information can be guaranteed by digital signatures, whereas the information confidentiality is achieved using encryption schemes. One can first sign and then encrypt a message when both authenticity and confidentially are desired. This approach is known as sign-then-encrypt. The main disadvantage of this solution is that it expands the final ciphertext's size and increase the sender and receiver's computing time, as signing and encryption are preformed in two separate steps. This motivated Zheng [21] to propose a cryptographic primitive signcryption. The idea of this kind of primitive is to perform encryption and signature in a single logical step in order to obtain confidentiality and authentication more efficiently than the sign-then-encrypt approach. Based on discrete algorithm problem, signcryption costs 58% less in average computation time and 70% less in message expansion than sign-then-encrypt does. Using RSA cryptosystem, it costs on average 50% less in computation time and 91% less in message expansion than sign-then-encrypt does. After the introduction of signcryption, many efficient signcryption schemes have been proposed [2, 18, 11, 6, 4, 20, 10, 12, 15].

Earlier signcryption schemes were only considered in traditional public key cryptography, where there is a certificate authority (CA) who generates certificates to bind a user with its public key. History has shown that the certificates in traditional PKI are costly to use and manage. Shamir [16] introduced the notion of Identity-based (or ID-based) cryptography to easy the above problem. In the new setting, the user's public key is some unique information about the identity of the user (e.g., a user's email address) which is assumed to be publicly known. The ability to use identities as public keys avoids the need to distribute public key certificates. This can be very useful in applications such as email where the recipient is often off-line and unable to present a public-key certificate while the sender encrypts a message. In ID-based system, a trusted third party, called the Private Key Generator (PKG), generates users' private keys. The PKG first publishes a master public key, and retains the corresponding master secret key. To obtain a private key, one should contact PKG, which uses the master secret key to generate the corresponding private key. Encryption or verification in ID-based cryptography only needs PKG's master public key and the user's identity information. In [16], Shamir proposed a concrete ID-based signature (IBS) scheme, but the construction of Identity-based encryption (IBE) remained as an open problem until the first efficient and fully functional identity-based encryption scheme proposed in [3]. This construction is built from a bilinear map (for example, the Weil pairing on elliptic curves). After that, identity-based cryptographic protocols from pairings have been extensively investigated by researchers [13, 5].

The first identity-based signcryption scheme was proposed in [11]. In this construction, the signature of the plaintext is visible in the ciphertext and thus, does not satisfy the semantic security. This flaw was fixed by Libert and Quisquater [10] by proposing a new construction. Boyen [4] proposed a multipurpose identity-based signcryption and formally defined the security notions of signcryption in identity-based cryptography. After that, Chen

978-0-7695-3354-4/08 $25.00 © 2008 IEEE
DOI 10.1109/NPC.2008.12

and Malone-Lee [6] proposed a more efficient scheme in the model defined in [4].

The notion of on-line/off-line signature was introduced by Even, Goldreich, and Micali [8]. The idea is to divide the signature generating procedure by two phases. The first phase is performed off-line (before the message to be signed is known) and the second phase is performed on-line (after the message to be signed is given). On-line/off-line signature schemes are useful, as in many applications the signer has a very limited response time once the message is presented, but he can carry out costly computations between consecutive signing requests. On-line/off-line signature schemes are particularly useful in smart card applications: The off-line phase is implemented either during the card manufacturing process or as a background computation whenever the card is connected to power, and the on-line phase uses the stored result of the off-line phase to sign actual messages. The on-line phase is typically very fast, and hence can be executed efficiently even on a weak processor.

Some signature schemes can be naturally partitioned into on-line and off-line phases. For example, the first step in the Fiat-Shamir, Schnorr, El-Gamal and DSS signature schemes does not depend on the given message, and can thus be carried out off-line. Even, Goldreich, and Micali [8] proposed the first generic method to convert any signature scheme into an on-line/off-line one. Their construction is not efficient as it increases the length of each signature by a quadratic factor. In 2001, Shamir and Tauman proposed another generic method to achieve on-line/off-line signing [17]. They use the notion of a trapdoor hash function to develop a paradigm called "hash-sign-switch", which can convert any signature scheme into a highly efficient on-line/off-line signature scheme. The on-line signing phase of their scheme maintains the efficiency of Even, Goldreich and Micali's scheme(requiring only one hash function), but the size of each signature increases only by a factor of two. Chen et al [7] proposed a much more efficient generic on-line/off-line signature scheme. Compared with Shamir-Tauman's signature scheme, their scheme has the advantages of the lower computation and storage cost for the off-line phase, and the lower communication cost for the on-line phase.

The notion of on-line/off-line signcryption was introduced by An, Dodis, and Tabin [1]. They did not give any concrete method in their work but general security proofs on signcryption schemes. They gave the security analysis of "encrypt-then-sign", "sign-then-encrypt" and "commit-then-encrypt-and-sign" under both insider and outsider attack models. The latter method can be combined with the "hash-sign-switch" technique to produce a generic on-line/off-line signcryption. The first practical on-line/off-line signcryption was proposed by Zhang, Mu, and Susilo in 2005 [20]. Their scheme is efficient as the on-line part does not require any exponent computations. They also employed the notion of short signatures, which contributes to the short signature length of the on-line signature part.

Motivation and Contribution

To date, there is no construction of identity-based on-line/off-line signcryption protocol in the literature. However, it would be of great practical interest to design an identity-based on-line/off-line signcryption. As it avoids the need to distribute public key certificates, identity-based cryptography has found many advantages in the systems as Adhoc networks, Mobile networks, etc. However, entities in these systems are normally less powerful than their counterparts such as desktops. This limits their ability to perform public key operations as encryption and signing. It will be certainly desirable if the above operations can be done in an efficient manner and, entities are able to perform some of operations beforehand. All these desirable properties can be achieved in identity-based on-line/off-line signcryption.

Our contributions of this paper are twofold. We first formally define the identity-based on-line/off-line signcryption and its security models. We specify two security notions, namely ciphertext indistinguishable and existentially unforgeable, in identity-based on-line/off-line signcryption. Both two notions capture the practical requirements of identity-based on-line/off-line signcryption. We then propose an ID-based on-line/off-line signcryption. Our construction is based on pairing on elliptic curves. It can achieve authenticity and confidentiality simultaneously in an efficient manner. All costly operations are performed in the off-line phase. The on-line part does not require any operations in the pairing group $\mathbb{G}$, and only includes one symmetric key encryption and the addition operations modular q, where q is a prime and its length depends on the system security parameter. We give a rigorous proof to show that our scheme is ciphertext indistinguishable under decisional Bilinear Diffie-Hellman assumption and is existentially unforgeable under computational Diffie-Hellman assumption. We finally show the potential application of our scheme in secure communications in wireless sensor network (WSN).

The rest of this paper is organized as follows. Next section briefly reviews the preliminaries required in this paper. In Section 3, we formally define the identity-based on-line/off-line signcryption. We present our scheme and prove its security in our model in Section 4 and Section 5. Section 6 shows the potential applications of our scheme. We conclude this paper in Section 7.

2. Preliminaries

Before presenting our results we briefly review the definition for groups equipped with a bilinear map, and the definitions of CDHP and DBDHP.

2.1. Bilinear Mapping

Let k be a security parameter and q be a k-*bit* prime number. Let us consider groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order q. For our purposes, we need a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfying the following properties:

1. Bilinearity: $\forall P, Q \in \mathbb{G}_1, \forall a, b \in Z_q^*$, $e(aP, bQ) = e(P, Q)^{ab}$.

2. Non-degeneracy: for any point $P \in \mathbb{G}_1, e(P, Q) = 1$ for all $Q \in \mathbb{G}_1$ iff $P = \mathcal{O}$.

3. Computability: there exists an efficient algorithm to compute $e(P, Q)$, for $P, Q \in \mathbb{G}_1$.

Such non-degenerate admissible maps over cyclic groups can be obtained from the Weil or the Tate pairing over supersingular elliptic curves [3] or abelian varieties [14].

2.2. Security Assumptions

The security of our scheme relies on the hardness of the following problems.

Definition 1. Computational Diffie-Hellman Problem (CDHP) *Given $(P, aP, bP) \in \mathbb{G}^3$ as the input, output abP.*

An algorithm $\mathcal{A}$ has advantage ϵ in solving CDHP in group $\mathbb{G}$ if $\Pr[\mathcal{A}(P, aP, bP) = abP] \geq \epsilon$, where the probability is over the random choices of (a, b), and the coin tosses of $\mathcal{A}$. We say an algorithm $\mathcal{A}$ (t, ϵ)-breaks CDHP in $\mathbb{G}$ if in time t, $\mathcal{A}$ has advantage ϵ in solving CDHP.

Definition 2. Decisional Bilinear Diffie-Hellman Problem (DBDHP) *Given two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of the same prime order q, a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, a generator P of $\mathbb{G}_1$, $(aP, bP, cP) \in {\mathbb{G}_1}^3$ and an element $h \in \mathbb{G}_2$, decide whether $h = e(P, P)^{abc}$ or not.*

An algorithm $\mathcal{D}$ has advantage ϵ in solving DBDHP in $(\mathbb{G}_1, \mathbb{G}_2)$ if $|\Pr_{a,b,c \in_R Z_q^*, h \in_R \mathbb{G}_2}[1 \leftarrow \mathcal{D}(P, aP, bP, cP, h)] - \Pr_{a,b,c \in_R Z_q^*}[1 \leftarrow \mathcal{D}(P, aP, bP, cP, e(P, P)^{abc})]| \geq \epsilon$. We say an algorithm $\mathcal{D}$ (t, ϵ)-breaks DBDHP in $(\mathbb{G}_1, \mathbb{G}_2)$ if in time t, $\mathcal{D}$ has advantage ϵ in solving DBDHP.

3. Syntax and Security Models of Identity-based On-line/Off-line Signcryption

We define the syntax and the security models of identity-based on-line/off-line signcryption.

3.1. Syntax of ID-based On-line/Off-line Signcryption

Definition 3. *ID-based on-line/off-line signcryption scheme is comprised of five algorithms:* **Setup**, **Extract**, **OffSign**, **OnSigncrypt** *and* **UnSigncrypt**.

1. **Setup**$(k) \rightarrow (params, s)$. Given a security parameter k as input, the private key generator (PKG) generates the system's public parameters $params$ and the master secret key s, where $params$ is published in the system and s is kept as secret by PKG.

2. **Extract**$(params, ID, s) \rightarrow d_{ID}$. Given an identity ID and the master secret key s as input, the PKG computes the corresponding private key d_{ID} and transmits it to its owner in a secure way.

3. **OffSign**$(params, ID_S, ID_R, d_{ID_S}) \rightarrow \sigma'$. Given $params$, ID_s's secret key d_{ID_S} and the receiver's identity ID_R as input, this algorithm outputs an off-line signature σ'.

4. **OnSigncrypt**$(params, m, ID_R, \sigma') \rightarrow C$. Given a message m, receiver's identity ID_R and an off-line signature σ' as input, this algorithm outputs the ciphertext C.

5. **UnSigncrypt**$(params, C, ID_S, ID_R, d_{ID_R}) \rightarrow \{m, \perp\}$. Given $params$, a ciphertext C, the sender's identity ID_S and the receiver's secret key d_{ID_R} as input, this algorithm outputs the plaintext m or the symbol "$\perp$". "$\perp$" denotes that C is an invalid ciphertext between ID_S and ID_R.

For simplicity, we omit the notation of $params$ from the inputs of **OffSign**, **OnSigncrypt** and **UnSigncrypt** in the rest of this paper.

Correctness. The algorithm **UnSigncrypt** will output a plaintext if the ciphertext and the off-line signature are generated as defined above.

$m \leftarrow$ **UnSigncrypt**$(params,$ **OnSigncrypt**$(params, m, ID_R,$ **OffSign**$(params, ID_S, ID_R, d_{ID_S})), ID_S, ID_R, d_{ID_R})$

3.2. Security Models of Identity-based On-line/Off-line Signcryption

We now state the security of identity-based on-line/off-line signcryption.

The first security notion is the ciphertext indistinguishability against adaptive chosen-ciphertext attacks. It is defined by the game as follows.

Definition 4. *We say that an identity-based on-line/off-line signcryption scheme (IDSC) has the* ***ciphertext indistinguishability against adaptive chosen-ciphertext attacks*** *property (IND-IDSC-CCA2) if no polynomially bounded adversary has a non-negligible advantage in the following game.*

1. The challenger runs the **Setup** algorithm with a security parameter k and sends the system parameters $params$ to the adversary $\mathcal{A}$.

2. The adversary $\mathcal{A}$ performs a polynomially bounded number of requests:

   (a) **Signcryption request**: $\mathcal{A}$ produces two identities ID_i, ID_j and a plaintext m. The challenger first computes ID_i's secret key d_{ID_i} = **Extract**(ID_i, s). Then, it runs the algorithm **OffSign**$(params, ID_i, ID_j, d_{ID_i})$ to obtain an off-line signature σ'. Finally, it returns **OnSigncrypt**$(m, \sigma', d_{ID_i}, ID_j)$ to $\mathcal{A}$.

   (b) **UnSigncryption request**: $\mathcal{A}$ produces two identities ID_i and ID_j, a ciphertext C. The challenger generates the private key d_{ID_j} = **Extract**(ID_j) and sends the result of **UnSigncrypt**(C, d_{ID_j}, ID_i) to $\mathcal{A}$ (this result could be the $\perp$ symbol if C is an invalid ciphertext).

   (c) **Key extraction request**: $\mathcal{A}$ produces an identity ID and receives the extracted private key d_{ID} = **Extract**(ID, s).

   $\mathcal{A}$ can present its requests adaptively: every request may depend on the answers to the previous ones.

3. $\mathcal{A}$ chooses two plaintexts m_0, m_1 in the message space specified in $params$ and two identities ID_A and ID_B on which he wishes to be challenged. The restriction is that $\mathcal{A}$ cannot choose ID_A or ID_B as a one of Key extraction requests.

4. The challenger takes a random bit $b \in_R \{0, 1\}$ and generates the ciphertext C^* for m_b as he responds the signcryption request.

5. $\mathcal{A}$ asks again a polynomially bounded number of requests just like in step 2. This time, he can not make a key extraction request on ID_A or ID_B and he cannot make an unSigncrypt query of (ID_A, ID_B, C^*).

6. Finally, A produces a bit b' and wins the game if $b' = b$.

The adversary's advantage is defined to be $Adv(\mathcal{A}) = |2\Pr[b' = b] - 1|$.

Definition 5. *An ID-based on-line/off-line signcryption scheme (IDSC) is said to be* ***existentially unforgeable against adaptive chosen-message attacks*** *(EF-IDSC-ACMA) if no polynomially bounded adversary has a non-negligible advantage in the following game.*

1. The challenger runs the **Setup** algorithm with a security parameter k and gives the system parameters $params$ to the adversary $\mathcal{A}$.

2. The adversary $\mathcal{A}$ performs a polynomially bounded number of requests as same as Def. 4.

3. Finally, $\mathcal{A}$ produces a triple (C^*, ID_A, ID_B). The restrictions are (C^*, ID_A, ID_B) is not the response of $\mathcal{A}$'s signcryption requests and ID_A has not been chosen as one of the key extract queries.

$\mathcal{A}$ wins the game if **Unsigncrypt**$(C^*, d_{ID_B}, ID_A) \neq \perp$. The adversary's advantage is simply its success probability $Adv(\mathcal{A}) = P[\mathcal{A}$ wins$]$.

Remarks. In Def. 5, the adversary is allowed to ask the private key corresponding to the identity ID_B in the challenging trip (C^*, ID_A, ID_B). This prevent a dishonest recipient ID_B to send a ciphertext to himself on behalf of ID_A and to try to convince a third party that ID_A was the sender.

4. The Scheme

In this section, we present our ID-based on-line/off-line signcryption scheme that satisfies the model introduced in Section 3. Assume that Alice and Bob are the sender and the receiver, respectively. The protocol is described as follows.

Setup: Given security parameters k, n and $\mathbb{G}_1, \mathbb{G}_2$ of order q and generator P of $\mathbb{G}_1$, picks a random $s \in_R Z_q^*$, and sets $P_{pub} = sP$. Chooses cryptographic hash functions $H_0 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_1 : \{0,1\}^* \times \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow Z_q^*$, $H_2 : Z_q^* \rightarrow \{0,1\}^n$ and $H_3 : \mathbb{G}_2 \rightarrow Z_q^* \times Z_q^*$. The system parameters are $(P, P_{pub}, H_0, H_1, H_2, H_3)$. The master key is s. H_0, H_1, H_2 and H_3 will be regarded as random oracles in security analysis.

Extract: Given an identity ID, the algorithm computes $d_{ID} = sH_0(ID)$ and outputs it as the private key related to ID corresponding to $Q_{ID} = H_0(ID)$.

OffSign: To send a message m to Bob, Alice follows the steps below. (1) Computes $Q_{ID_B} = H_0(ID_B)$. (2) Picks random $x, y \in Z_q^*$, and sets $k = H_3(e(P_{pub}, Q_{ID_B})^x)$. (3) Splits k into k_1, k_2 such that $k_1 \in Z_q^*$ and $k_2 \in Z_q^*$, then stores them for future use. (4) Given a secret key d_{ID_A}, outputs the off-line signature (S, U), where $S = d_{ID_A} - xP_{pub}$, $U = (y - k_1)P$; also stores x, y for future use.

OnSigncrypt: Given a message $m \in Z_q^*$ and an off-line signature (S, U), Alice sets $k_3 = H_2(k_2)$ first. The message encryption is done with k_3 and a symmetric-key encryption algorithm E such as AES. The ciphertext is $c = E_{k_3}(m)$. Computes $r = H_1(c, S, U)$ and on-line signature $\sigma = rx + y$; returns ciphertext (c, S, U, σ).

UnSigncrypt: Given ciphertext (c, S, U, σ), (1) Computes $T = e(-S, Q_{ID_B})e(Q_{ID_A}, d_{ID_B})$. (2) Sets $k = H_3(T)$, then splits k into k_1, k_2. (3) Sets $k_3 = H_2(k_2)$ and decrypts the message $D_{k_3}(c) = m$. The correct verification requires to verify the equality $e(\sigma P_{pub} + rS, P) = e(U + k_1P + rQ_{ID_A}, P_{pub})$, where $r = H_1(c, S, U)$.

Correctness: The consistency is easy to verify by the bilinearity of the map as follows:

$$\begin{aligned}
&e(\sigma P_{pub} + rS, P)\\
&= e((rx + y)P_{pub} + r(d_{ID_A} - xP_{pub}), P)\\
&= e(rxP_{pub} + yP_{pub} + rsQ_{ID_A} - rxP_{pub}, P)\\
&= e(yP_{pub} + rsQ_{ID_A}, P)\\
&= e(U + k_1P + rQ_{ID_A}, P_{pub})
\end{aligned}$$

$$\begin{aligned}
&e(-S, Q_{ID_B})e(Q_{ID_A}, d_{ID_B})\\
&= e(-d_{ID_A} + xP_{pub}, Q_{ID_B})e(sQ_{ID_A}, Q_{ID_B})\\
&= e(-d_{ID_A} + xP_{pub} + d_{ID_A}, Q_{ID_B})\\
&= e(P_{pub}, Q_{ID_B})^x
\end{aligned}$$

Performance and size: the proposed algorithms satisfy the requirement of on-line/off-line signcryption as all expensive computations are done in the off-line phase. The on-line phase consists of only two hashings, one multiplication, and a symmetric-key encryption. The size of our signature part (c, S, U, σ) is $2\log_2 \rho + \log_2 q + 160$, in which ρ stands for the safe length of group $\mathbb{G}_1$.

5. Proofs of Security

We now provide the security analysis of our scheme. Our proof for IND-IDSC-CCA2 is inspired by the proof in [10].

Theorem 1. *In the random oracle model, we assume we have an IND-IDSC-CCA2 adversary called $\mathcal{A}$ that is able to distinguish ciphertexts during the game of definition 4 with an advantage ϵ when running in a time t and asking H_0, H_1, H_2, H_3, key extraction oracle, on-line/off-line signcrypt oracle and on-line/off-line unsigncrypt oracle q_0, q_1, q_2, q_3, q_e, q_s and q_u times respectively. Then, there exists a distinguisher $\mathcal{B}$ that can solve the Decisional Bilinear Diffie-Hellman problem in a time $O(t+(2q_s+2q_u(q_3+q_s+q_u))\mathcal{T})$ with an advantage*

$$Adv(\mathcal{B})^{DBDH(G_1,P)} > 2(\epsilon - (q_1 + q_s + q_u)/2^{k-1})/q_{H_0}^4$$

where $\mathcal{T}$ denotes the computation time of the bilinear map.

Proof. The distinguisher $\mathcal{B}$ receives a random instance (P, aP, bP, cP, h) of the Decisional Bilinear Diffie-Hellman problem. His goal is to decide whether $h = e(P, P)^{abc}$ or not. $\mathcal{B}$ will run $\mathcal{A}$ as a subroutine and act as $\mathcal{A}$'s challenger in the IND-IDSC-CCA2 game. $\mathcal{B}$ needs to maintain lists L_0, L_1, L_2 and L_3 that are initially empty and are used to keep track of answers to queries asked by $\mathcal{A}$ to oracles H_0, H_1, H_2 and H_3. We assume that any Signcrypt or Unsigncrypt request on a pair of identities happens after $\mathcal{A}$ asked the hashing H_0 of these identities. Any key extraction query on an identity is also preceded by a hash query on the same identity. We also assume $\mathcal{A}$ never makes an unsigncryption query on a ciphertext obtained from the signcryption oracle. He only makes unsigncryption queries for observed ciphertexts.

At the beginning of the game, $\mathcal{B}$ gives $\mathcal{A}$ the system parameters with $P_{pub} = cP$ (c is unknown to $\mathcal{B}$ and plays the role of the PKG's master key). Then, $\mathcal{B}$ chooses two distinct random numbers $i, j \in \{1, ..., q_{H_0}\}$. $\mathcal{A}$ asks a polynomially bounded number of H_0 requests on identities of his choice. At the i^{th} H_0 request, $\mathcal{B}$ answers by $H_0(ID_i) = aP$. At the j^{th}, he answers by $H_0(ID_j) = bP$. The private keys d_{ID_i} and d_{ID_j} (which are not computable by $\mathcal{B}$) are respectively acP and bcP. For requests $H_0(ID_e)$ with $e \neq i, j$, $\mathcal{B}$ chooses be $b_e \in_R Z_q^*$, puts the pair (ID_e, b_e) in list L_0 and answers $H_0(ID_e) = b_eP$.

We now explain how the other kinds of requests are treated by $\mathcal{B}$.

H_1 **requests**: for a query $H_1(c_e, S_e, U_e)$, $\mathcal{B}$ first ensures the list L_1 does not contain a tuple (c_e, S_e, U_e, r_e). If such a tuple is found, $\mathcal{B}$ answers r_e, otherwise he chooses $r \in_R Z_q^*$, gives it as an answer to the query and puts the tuple (c_e, S_e, U_e, r) into L_1.

H_2 **requests**: on a $H_2(k_{2_e})$ request, $\mathcal{B}$ searches a pair (k_{2_e}, k_{3_e}) in the list L_2. If such a pair is found, $\mathcal{B}$ answers by k_{3_e}, otherwise he answers $\mathcal{A}$ by a random binary sequence $k_3 \leftarrow_R \{0, 1\}^n$ such that no entry $(., k_3)$ exists in L_2 and puts the pair (k_{2_e}, k_3) into L_2.

H_3 **requests**: on a $H_3(g_e)$ request, $\mathcal{B}$ searches a pair (g_e, k_e) in the list L_3. If such a pair is found, $\mathcal{B}$ answers by k_e, otherwise he answers $\mathcal{A}$ by a random $k \leftarrow_R Z_q^*$ such that no entry $(., k)$ exists in L_3 and puts the pair (g_e, k) into L_3.

Key extraction requests: when $\mathcal{A}$ asks a query **Extract**(ID_A), if $ID_A = ID_i$ or $ID_A = ID_j$, then $\mathcal{B}$ fails and stops. If $ID_A \neq ID_i, ID_j$ then the list L_0 must contain a pair (ID_A, b_e) for some b_e (this indicates $\mathcal{B}$ previously answered $H_0(ID_A) = b_eP$ on a H_0 query on ID_A). The private key corresponding to ID_A is then $b_eP_{pub} = cb_eP$. It is computed by $\mathcal{B}$ and returned to $\mathcal{A}$.

Signcrypt requests: At any time $\mathcal{A}$ can perform a Signcrypt request for a plaintext m and identities ID_A and ID_B.

In the case $ID_A \neq ID_i, ID_j$, $\mathcal{B}$ computes the private key d_{ID_A} corresponding to ID_A by running the key extraction request algorithm and retrieves the (ID_B, b_e) to get public key corresponding to ID_B from L_0. $\mathcal{B}$ can simply run the **OffSign** and **OnSigncrypt** algorithms.

In the case $ID_A = ID_i$ or $ID_A = ID_j$ and $ID_B \neq ID_i, ID_j$, $\mathcal{B}$ has to simulate the execution of **OffSign** and **OnSigncrypt** algorithms. In **OffSign** phase: (1) Randomly chooses $y_e, r_e \in_R Z_q^*$. (2) Sets $S_e = y_e P_{pub}$ and $U_e = r_e P + r_e y_e P - r_e Q_{ID_A}$. (3) Computes $T_e = e(-S_e, Q_{ID_B}) e(Q_{ID_A}, d_{ID_B})$ where d_{ID_B} is the private key corresponding to ID_B ($\mathcal{B}$ could obtain it from the key extraction algorithm because $ID_B \neq ID_i, ID_j$). (4) Runs the H_3 simulation algorithm to find $k_e = H_3(T_e)$. (5) Splits k_e into k_{1_e} and k_{2_e}. In **OnSigncrypt** phase: (1) Runs the H_2 simulation algorithm to find $k_{3_e} = H_2(k_{2_e})$. (2) Computes $c_e = E_{k_{3_e}}(m)$. (3) Computes $\sigma_e = k_{1_e} + r_e$. (4) Puts (c_e, S_e, U_e, r_e) into L_1 and the ciphertext $(c_e, S_e, U_e, \sigma_e)$ is returned to $\mathcal{A}$.

If ID_A and ID_B are the identities ID_i and ID_j. $\mathcal{B}$ has to simulate the execution of **OffSign** and **OnSigncrypt** algorithms. In **OffSign** phase: (1) Randomly chooses $y_e, r_e^* \in_R Z_q^*$. (2) Sets $S_e^* = y_e P_{pub}$ and $U_e^* = r_e^* P + r_e^* y_e P - r_e^* Q_{ID_A}$. (3) Randomly chooses $T_e^* \in_R G_2$ and $k_e \in_R Z_q^*$ such that no entry $(., k_e)$ in L_3 and puts (T_e^*, k_e) in L_3. (4) Splits k_e into k_{1_e} and k_{2_e}. In **OnSigncrypt** phase: (1) Runs the H_2 simulation algorithm to find $k_{3_e} = H_2(k_{2_e})$. (2) Computes $c_e^* = E_{k_{3_e}}(m)$. (3) Computes $\sigma_e^* = k_{1_e} + r_e^*$. (4) Puts $(c_e^*, S_e^*, U_e^*, r_e^*)$ into L_1 and the ciphertext $(c_e^*, S_e^*, U_e^*, \sigma_e^*)$ is returned to $\mathcal{A}$.

Unsigncrypt requests : When receiving an unsigncryption query for a ciphertext $(c_e, S_e, U_e, \sigma_e)$ for identities ID_A and ID_B that are not ID_i and ID_j, $\mathcal{B}$ first checks if the list L_1 contains (c_e, S_e, U_e, r_e). If no such tuple is found, $\mathcal{B}$ rejects the ciphertext. Otherwise, he computes $T_e = e(-S_e, Q_{ID_B}) e(Q_{ID_A}, d_{ID_B})$ where d_{ID_B} is the private key corresponding to ID_B ($\mathcal{B}$ could obtain it from the key extraction algorithm because $ID_B \neq ID_i, ID_j$). He runs the H_3 simulation algorithm to find $k_e = H_3(T_e)$ and split k_e into k_{1_e}, k_{2_e}. $\mathcal{B}$ verifies if $e(\sigma_e P_{pub} + r_e S_e, P) = e(U_e + k_{1_e} P + r_e Q_{ID_A}, P_{pub})$, where $r_e = H_1(c_e, S_e, U_e)$. if not, he rejects the ciphertext. He then searches for a query $H_2(k_{2_e})$ in list L_2. If no such query is found, $\mathcal{B}$ takes a random pair $(k_{2_e}, k_{3_e}) \in Z_q^* \times \{0,1\}^n$ such that no $(., k_{3_e})$ already exists in L_2 and inserts (k_{2_e}, k_{3_e}) into L_2. He finally uses the corresponding k_{3_e} to find $m_e = D_{k_{3_e}}(c_e)$ and returns m_e. If no message has been returned, return $\perp$.

When $\mathcal{A}$ observes a ciphertext $(c_e, S_e, U_e, \sigma_e)$ for identities ID_i and ID_j, he may want to ask $\mathcal{B}$ for the unsigncryption of the ciphertext. $\mathcal{B}$ steps through the list L_3 with entries (T_e, k_e) as following: splits k_e into k_{1_e}, k_{2_e}. $\mathcal{B}$ verifies if $e(\sigma_e P_{pub} + r_e S_e, P) = e(U_e + k_{1_e} P + r_e Q_{ID_A}, P_{pub})$, where $r_e = H_1(c_e, S_e, U_e)$. if not, he moves to the next element in L_3 and begins again, else searches for a query $H_2(k_{2_e})$ in list L_2. If no such query is found, $\mathcal{B}$ takes a random pair $(k_{2_e}, k_{3_e}) \in Z_q^* \times \{0,1\}^n$ such that no $(., k_{3_e})$ already exists in L_2 and inserts (k_{2_e}, k_{3_e}) into L_2. He finally uses the corresponding k_{3_e} to find $m_e = D_{k_{3_e}}(c_e)$ and returns m_e. If no message has been returned, return $\perp$. If $\mathcal{A}$ previously asked the hash value $H_1(c_e, S_e, U_e)$, there is a probability of at most $1/2^k$ that $\mathcal{B}$ answered r_e. The simulation fails if L_1 contains a tuple (c_e, S_e, U_e, r_e). We can find that the probability to reject a valid ciphertext does not exceed $q_u/2^k$.

After a polynomially bounded number of queries, $\mathcal{A}$ chooses a pair of identities on which he wishes to be challenged. With a probability at least $1/C^2_{q_{H_0}}$ this pair of target identities will be (ID_i, ID_j). If $\mathcal{A}$ asks the private key of ID_i or ID_j before choosing his target identities, then B fails because he is unable to answer the question. If $\mathcal{A}$ actually chooses to be challenged on ID_i and ID_j , then he cannot ask ID_i nor ID_j 's private keys in the second stage. If $\mathcal{A}$ does not choose ID_i and ID_j as target identities, then $\mathcal{B}$ fails.

When $\mathcal{A}$ produces his two plaintexts m_0 and m_1, $\mathcal{B}$ chooses a random bit $b \in_R \{0,1\}$ and signcrypts m_b. To do so, $\mathcal{B}$ follows the steps below. (1) Randomly chooses $y_e, r_e^* \in_R Z_q^*$. (2) Sets $S_e^* = y_e P_{pub}$ and $U_e^* = r_e^* P + r_e^* y_e P - r_e^* Q_{ID_A}$. (3) Computes $T_e^* = e(-S_e^*, Q_{ID_B}) h$(where h is $\mathcal{B}$'s candidate for the DBDH problem). (4) Runs the H_3 simulation algorithm to find $k_e = H_3(T_e^*)$ and split k_e into k_{1_e},k_{2_e}. (5) Sets $k_{3_e} = H_2(k_{2_e})$(H_2 is the simulator) and computes $c_b^* = E_{k_{3_e}}(m_b)$. (6) Computes $\sigma_e^* = k_{1_e} + r_e^*$. (7) Verifies as above if L_1 already contains an entry (c_b^*, S_e^*, U_e^*, r) such that $r \neq r_e^*$. If not, he puts the tuple $(c_b^*, S_e^*, U_e^*, r_e^*)$ into L_1. In the opposite case, $\mathcal{B}$ repeats the process until finding a tuple $(c_b^*, S_e^*, U_e^*, r_e^*)$ whose first three elements do not figure in an entry of L_1. Once he has admissible elements $(S_e^*, U_e^*, \sigma_e^*, r_e^*)$. $\mathcal{B}$ just has to send the ciphertext $(c_b^*, S_e^*, U_e^*, \sigma_e^*)$ to $\mathcal{A}$.

$\mathcal{A}$ then performs a second series of queries which is treated in the same way as the first one. At the end of the simulation, he produces a bit b' for which he believes the relation $ciphertext = Signcrypt(m_{b'}, d_{ID_i}, ID_j)$ holds. At this moment, if $b = b'$, $\mathcal{B}$ then answers 1 as a result because his candidate h allowed him to produce a ciphertext that appeared to $\mathcal{A}$ as a valid signcrypted text of m_b. If $b \neq b'$, B then answers 0.

Let us now consider how our simulation could fail i.e. describe events that could cause $\mathcal{A}$'s view to differ when run by $\mathcal{B}$ from its view in a real attack. It is clear that the simulations for H_0, H_1, H_2 and H_3 are indistinguishable from real random oracles. Because errors of On-line signcrypt is a consequence of off-line signcrypt. We analyze them together. The only possibilities for introducing an error here are defining $H_1(c_e, S_e, U_e)$ when it is already defined. Since S_e and U_e take their values uniformly at random in G_1, the chance of one of these events occurring is at most $(q_1 + q_s)/2^k$ for each query. The probability for unsigncrypt simulator to reject a valid ciphertext does not exceed $q_u/2^k$ as mentioned before. We saw that $\mathcal{B}$ fails if $\mathcal{A}$ asks the private key associated to ID_i or ID_j during the first stage. We know that there are $C^2_{q_{H_0}}$ ways to choose the pair (ID_i, ID_j). Among those $C^2_{q_{H_0}}$ pairs of identities, at least one of them will never be the subject of a key extraction query from $\mathcal{A}$. Then, with a probability greater than $1/C^2_{q_{H_0}}$ $\mathcal{A}$ will not ask the questions $Keygen(ID_i)$ and $Keygen(ID_j)$. Further, with a probability exactly $1/C^2_{q_{H_0}}$ $\mathcal{A}$ chooses to be challenged on the pair (ID_i, ID_j) and this must allow $\mathcal{B}$ to solve his decisional problem if $\mathcal{A}$ wins the IND-IDSC-CCA game.

Since

$$\begin{aligned} p_1 &= P[b' = b|\sigma = Signcrypt(m_b, d_{ID_i}, ID_j)] \\ &= (\epsilon + 1)/2 - (q_1 + q_s + q_u)/2^k \end{aligned}$$

$$p_0 = P[b' = i|h \in_R G_2] = 1/2 (i = 0, 1)$$

We then have

$$\begin{aligned} Adv[\mathcal{B}] &= |P_{a,b,c \in_R Z_q^*}[1 \leftarrow \mathcal{B}(P, aP, bP, cP, e(P,P)^{abc})] \\ &\quad - P_{a,b,c \in_R Z_q^*, h \in_R G_2}[1 \leftarrow \mathcal{B}(P, aP, bP, cP, h)]| \\ &= \frac{|p_1 - p_0|}{(C^2_{q_{H_0}})^2} = \frac{\epsilon - (q_1 + q_s + q_u)/2^{k-1}}{2(C^2_{q_{H_0}})^2} \\ &> 2(\epsilon - (q_1 + q_s + q_u)/2^{k-1})/q^4_{H_0} \end{aligned}$$

□

The unforgeability against adaptive chosen messages attacks [9], defined in Definition 5, derives from the security of the scheme in [19], under the computational Diffie-Hellman assumption. Due to space limitation, we omit the proof in this version of the paper. One can show that an attacker that is able to forge a signcrypted message must be able to forge a signature for the scheme in [19].

6. Application

A wireless sensor network(WSN) is composed of a large number of sensor nodes that are densely deployed either inside the phenomenon or very close to it. A sensor node will communicate with other nodes or a destination (sink) node frequently. PKI is not suitable for WSN secure communication because of the certificate overhead. Our ID-based Scheme may be applied for sending encrypted data in WSN, there is no need to bind a public key to its owners identity since those are one single thing. Because of limited computation power and network bandwidth in WSN, some efficient security algorithms for secure mobile communications are needed. Our Scheme achieves both ciphertext size efficiency and computation efficiency, so it's a good candidate for WSN secure communication.

7. Conclusion

In this paper, we have proposed an ID-based on-line/off-line signcryption scheme. In our scheme, the on-line computation is very efficient. Our scheme is proved secure against existential forgery under adaptive chosen message attacks based on the random oracle model assuming that CDH problem is hard, and it's also secure against adaptive chosen ciphertext attacks under the notion of indistinguishability of ciphertext on the random oracle model assuming that DBDH problem is hard. We also give some application. The scheme may be suitable for WSN secure communication, because of the computation and ciphertext size efficiency. Our future work involves proposing a generic ID-based on-line/off-line signcryption scheme.

References

[1] J. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Advances in Cryptology - Eurocrypt02*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer-Verlag, 2002.

[2] F. Bao and R. H. Deng. A signcryption scheme with signature directly verifiable by public key. In *1st International Workshop on Practice and Theory in Public Key Cryptography (PKC1998)*, volume 1431 of *Lecture Notes In Computer Science*, pages 55–59. Springer, 1998.

[3] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Advances in Cryptology - Crypto01*, volume 2139 of *LNCS*. Springer, 2001.

[4] X. Boyen. Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography. In *Advances in Cryptology - Crypto2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 383–399. Springer-Verlag, 2003.

[5] J. C. Cha and J. H. Cheon. An identity-based signature from gap diffie-hellman groups. In *Practice and Theory in Public Key Cryptography - PKC2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 18–30, Miami, USA, 2003. Springer-Verlag.

[6] L. Chen and J. Malone-Lee. Improved identity-based signcryption. Cryptology ePrint Archive, Report 2004/114, 2004. http://eprint.iacr.org.

[7] X. Chen, F. Zhang, W. Susilo, and Y. Mu. Efficient generic on-line/off-line signatures without key exposure. In *International Conference on Applied Cryptography and Network Security(ACNS 2007)*, volume 4521 of *Lecture Notes in Computer Science*, pages 18–30, Zhuhai, China, 2007. Springer-Verlag.

[8] S. Even, O. Goldreich, and S. Macali. On-line/off-line digital signatures. In *Proceedings of Advances in Cryptology: Crypto 89*, 1990.

[9] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen message attack. *SIAM J. Comp*, 17(2):281–308, 1988.

[10] B. Libert and J.-J. Quisquater. New identity-based signcryption schemes based on pairings. In *IEEE Information Theory Workshop*, Paris, France, 2003.

[11] J. Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002. http://eprint.iacr.org.

[12] D. Nalla and K. C. Reddy. Signcryption scheme for identity-based cryptosystems. Cryptology ePrint Archive, Report 2003/066, 2003. http://eprint.iacr.org.

[13] K. Paterson. Id-based signatures from pairings on elliptic curves. Cryptology ePrint Archive, Report 2002/004, 2002. http://eprint.iacr.org.

[14] K. Rubin and A. Silverberg. The best and worst of super-singular abelian varieties in cryptology. Cryptology ePrint Archive, Report 2002/006, 2002. http://eprint.iacr.org.

[15] R. Sakai and M. Kasahara. Id-based cryptosystems with pairing on elliptic curve. In *Symposium on Cryptography and Information Security – SCIS'2003*, Hamamatsu, Japan, 2003.

[16] A. Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology C Crypto84*, volume 0196 of *Lecture Notes in Computer Science*, pages 47–53. Springer-Verlag, 1984.

[17] A. Shamir and Y. Tauman. Improved online/offline signature schemes. In *Advances in Cryptology-Crypto*, volume 2139 of *Lecture Notes in Computer Science*, pages 355–367. Springer-Verlag, 2001.

[18] R. Steinfeld and Y. Zheng. A signcryption scheme based on integer factorization. In *Proceedings of Information Security Workshop 2000 (ISW2000)*, volume 1975 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 2000.

[19] S.Xu, Y.Mu, W.Susilo, X.Chen, X.Huang, and F. Zhang. Online/offline signatures and multisignatures for aodv and dsr routing security. Cryptology ePrint Archive, Report 2006/236, 2006. http://eprint.iacr.org.

[20] F. Zhang, Y. Mu, and W. Susilo. Reducing security overhead for mobile networks. In *Advanced Information Networking and Applications AINA 2005*, volume 1, pages 398–403, 2005.

[21] Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption). In *Advances in Cryptology – Crypto'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 165–179. Springer-Verlag, 1997.