An Efficient Identity-Based Online/Offline Encryption Scheme*

Joseph K. Liu and Jianying Zhou

Institute for Infocomm Research
Singapore
{ksliu,jyzhou}@i2r.a-star.edu.sg

Abstract. In this paper, we present an efficient Identity-based Online / Offline Encryption (IBOOE) scheme. An IBOOE scheme allows one to split the encryption into two phases. In the offline phase, most heavy computations such as exponentiation or pairing, if any, are done in this phase. Yet it does not require the knowledge of the plaintext or the receiver's identity. This nice property allows it can be executed 'offline', or inside some powerful device. The next phase is called the online phase, where only light computations such as integer addition, multiplication or hashing are needed, together with the plaintext and the receiver's identity. This can be executed inside some embedded device such as smart card or wireless sensor where the computation power is very limited. We propose an efficient IBOOE scheme, with great improvement in the computation requirement of both the offline, online encryption phase and decryption phase, together with much shorten ciphertext over previous schemes. Our scheme can be proven secure in the random oracle model.

1 Introduction

The notion of "online / offline" cryptographic algorithm was first introduced by Even, Goldreich and Micali [5], in the context of digital signature. With this notion, the signing process can be divided into two phases. The first phase is called *offline* phase which is executed prior to the arrival of a message and the second phase is called *online* phase which is performed after knowing the message. The online phase should be very fast and require only very light computation, such as integer multiplication or hashing. Other heavier computation such as exponentiation should be avoided in the online phase. In this way, online / offline schemes are particularly useful for low-power devices such as smartcard or wireless sensor applications. Those heavy computations are done in the offline phase which can be carried out by other powerful devices.

In parallel to online/offline signature [10,8,4,7], the first online/offline encryption scheme was first proposed by Guo, Mu and Chen [6]. Note that there is a slight difference in the definition between online/offline signature and encryption scheme. If we split the encryption process in the same way as the signing process

* The work in this paper is funded by the A*STAR project SEDS-0721330047.

M. Abdalla et al. (Eds.): ACNS 2009, LNCS 5536, pp. 156–167, 2009.

(that is, put all heavy computation into the offline phase), it is trivial to separate some standard encryption, such as ElGamal encryption scheme. However, it is only suitable for the situation where the sender knows the recipient of the encrypted message in the offline phase, since the offline phase requires the knowledge of the public key of the recipient. We are not interested in this scenario. Instead, we consider a notion that allows the knowledge of the recipient is yet unknown in the offline phase. [6] uses this definition for their scheme, in the context of identity-based encryption.

There are some scenarios that may require the above online/offline encryption. Suppose there are some sensitive data stored in a smartcard, which has only very limited computation power. In order to send the sensitive data to a recipient in a secure way, it should be encrypted using the recipient's public key or identity. To ensure timely and efficient delivery, it would be much better if part of the encryption process could be done *prior* to knowing the data to be encrypted *and* the recipient's public key or identity.

Wireless sensor network (WSN) can be another situation where online/offline encryption is useful. Similar to smartcard, wireless sensor also has only limited resource. It may take very long time, or even impossible to execute heavy computation. Yet the data they collect may be sensitive which is necessary to be encrypted before sending back to the base stations. By using online/offline encryption, the offline part (containing all heavy computation) can be done by a third party at the setup or manufacturing stage. Obviously at this stage nothing is collected. Sometimes even the base station identity maybe still unknown to the wireless sensor. Online/offline encryption is a good solution in this scenario.

Identity-Based (ID-Based) Cryptosystem, introduced by Shamir [9], eliminates the necessity for checking the validity of certificates in traditional public key infrastructure (PKI). In an ID-based cryptosystem, public key of each user is easily computable from an arbitrary string corresponding to this user's identity (e.g. an email address, a telephone number, etc.). Using its master key, a private key generator (PKG) then computes a private key for each user. This property avoids the requirement of using certificates and associates implicitly a public key (i.e. user identity) to each user within the system. One only needs to know the recipient's identity in order to send an encrypted message to him. It avoids the complicated and costly certificate (chain) verification for the authentication purpose. In contrast, the traditional PKI needs an additional certification verification process, which is equivalent to the computation of *two* signature verifications.

Identity-based system is particularly suitable for power constrained device such as WSN or smartcard. The absence of certificate eliminates the costly certificate verification process. In addition, when there is a new node added to the network, other nodes do not need to have its certificate verified in order to communicate in a secure and authenticated way. This can greatly reduce communication overhead and computation cost.

1.1 Contribution

In this paper, we propose an efficient identity-based online/offline encryption (IBOOE) scheme. There are only two IBOOE schemes existed in the literature,

both are proposed by Guo, Mu and Chen in [6]. Although they satisfy the definitions and basic requirements of an IBOOE, they are not actually very efficient. The first scheme (denoted by GMC-1) requires 7 pairing operations in the decryption stage. While for the second scheme (denoted by GMC-2), the ciphertext is very large (more than 6400 bits). Our proposed scheme provides a much better efficiency: We just require 2 pairing operations in the decryption stage. The ciphertext is only 1280 bits which is 40% smaller than GMC-1 and 80% smaller than GMC-2. Besides, our scheme requires lighter computation in both offline and online stage than both GMC-1 and GMC-2. Our scheme can be proven secure in the random oracle model.

1.2 Organization

The rest of our paper is organized as follow. Some definitions will be given in Section 2. We present our scheme in Section 3. It is followed by the detail comparison between our scheme and other schemes in Section 4. Finally we conclude the paper in Section 5.

2 Definitions

2.1 Pairings

We briefly review the bilinear pairing. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order q. Let P be a generator of $\mathbb{G}$, and e be a bilinear map such that $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ with the following properties:

1. *Bilinearity*: For all $U, V \in \mathbb{G}$, and $a, b \in \mathbb{Z}$, $e(aU, bV) = e(U, V)^{ab}$.
2. *Non-degeneracy*: $e(P, P) \neq 1$.
3. *Computability*: It is efficient to compute $e(U, V)$ for all $U, V \in \mathbb{G}$.

2.2 Intractability Assumption

Definition 1 (ℓ-Bilinear Diffie-Hellman Inversion Assumption (ℓ-BDHI)). *[2] The ℓ- Diffie-Hellman (ℓ-BDHI) problem in $\mathbb{G}$ is defined as follow: On input a $(\ell+1)$-tuple $(P, \alpha P, \alpha^2 P, \cdots, \alpha^\ell P) \in \mathbb{G}^{\ell+1}$, output $e(P, P)^{\frac{1}{\alpha}} \in \mathbb{G}_T$. We say that the (t, ϵ, ℓ)-BDHI assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ in solving the ℓ-BDHI problem in $\mathbb{G}$.*

2.3 Definition of ID-Based Online/Offline Encryption

An ID-based online/offline encryption scheme consists of the following five probabilistic polynomial time (PPT) algorithms:

– $(param, msk) \leftarrow \mathsf{Setup}(1^k)$ takes a security parameter $k \in \mathbb{N}$ and generates $param$ the global public parameters and msk the master secret key of the PKG.

- $D_{ID} \leftarrow \mathsf{Extract}(1^k, param, msk, ID)$ takes a security parameter k, a global parameters $param$, a master secret key msk and an identity ID to generate a secret key D_{ID} corresponding to this identity.
- $\bar{\phi} \leftarrow \mathsf{Offline\text{-}encrypt}(1^k, param)$ takes a security parameter k and a global parameters $param$ to generate an offline ciphertext $\bar{\phi}$.
- $\phi \leftarrow \mathsf{Online\text{-}encrypt}(1^k, param, m, \bar{\phi}, ID)$ takes a security parameter k, a global parameters $param$, a message m, an offline ciphertext $\bar{\phi}$, an identity ID to generate a ciphertext ϕ.
- $m/\perp \leftarrow \mathsf{Decrypt}(1^k, param, \phi, D_{ID})$ takes a security parameter k, a global parameters $param$, a ciphertext ϕ, a secret key of the receiver D_{ID} to generate a message m or $\perp$ which indicates the failure of decryption.

For simplicity, we omit the notation of 1^k and $param$ from the input arguments of the above algorithms in the rest of this paper.

2.4 Security of ID-Based Online/Offline Encryption

Definition 2 (Chosen Ciphertext Security). *An ID-based online/offline encryption scheme is semantically secure against chosen ciphertext insider attack (ID-IND-CCA) if no PPT adversary has a non-negligible advantage in the following game:*

1. *The challenger runs* Setup *and gives the resulting param to adversary $\mathcal{A}$. It keeps msk secret.*
2. *In the first stage, $\mathcal{A}$ makes a number of queries to the following oracles which are simulated by the challenger:*
   (a) ***Extraction oracle**: $\mathcal{A}$ submits an identity ID to the extraction oracle for the result of* Extract(msk, ID).
   (b) ***Decryption oracle**: $\mathcal{A}$ submits a ciphertext ϕ and a receiver identity ID to the oracle for the result of* Decrypt(ϕ, D_{ID}). *The result is made of a message if the decryption is successful. Otherwise, a symbol $\perp$ is returned for rejection.*

   These queries can be asked adaptively. That is, each query may depend on the answers of previous ones.
3. *$\mathcal{A}$ produces two messages m_0, m_1 and an identities ID^*. The challenger chooses a random bit $b \in \{0,1\}$ and computes an encrypted ciphertext $\phi^* =$* Online-Encrypt$(m_b,$ Offline-Encrypt$(), ID^*)$. *ϕ^* is sent to $\mathcal{A}$.*
4. *$\mathcal{A}$ makes a number of new queries as in the first stage with the restriction that it cannot query the decryption oracle with (ϕ^*, ID^*) and the extraction oracle with ID^*.*
5. *At the end of the game, $\mathcal{A}$ outputs a bit b' and wins if $b' = b$.*

$\mathcal{A}$'s advantage is defined as $\boldsymbol{Adv}^{IND-CCA}(\mathcal{A}) = |\Pr[b' = b] - \frac{1}{2}|$.

3 The Proposed Online/Offline ID-Based Encryption Scheme

3.1 Construction

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of prime-order q, and let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be the bilinear pairing. We use a multiplicative notation for the operation in $\mathbb{G}$ and $\mathbb{G}_T$.

Setup: The PKG selects a generator $P \in \mathbb{G}$ and randomly chooses $s, w \in_R \mathbb{Z}_q^*$. It sets $P_{pub} = sP$, $P'_{pub} = s^2P$ and $W = (w+s)^{-1}P$. Define $\mathcal{M}$ to be the message space. Let $n_M = |\mathcal{M}|$. Also let $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, $H_2 : \{0,1\}^* \times \mathbb{G}_T \rightarrow \mathbb{Z}_q^*$ and $H_3 : \{0,1\}^* \rightarrow \{0,1\}^{n_M}$ be some cryptographic hash functions. The public parameters *param* and master secret key *msk* are given by

$$param = (\mathbb{G}, \mathbb{G}_T, q, P, P_{pub}, P'_{pub}, W, w, \mathcal{M}, H_1, H_2, H_3) \qquad msk = s$$

Extract: To generate a secret key for a user with identity $ID \in \{0,1\}^{n_d}$, the PKG computes:

$$D_{ID} = (H_1(ID) + s)^{-1}P$$

Offline-Encrypt: Randomly generates $u, x, \alpha, \beta, \gamma, \delta \in_R \mathbb{Z}_q^*$ and computes:

$$U \leftarrow W - uP \qquad R \leftarrow e(wP + P_{pub}, P)^x$$

$$T_0 \leftarrow x\Big(w\alpha P + \big(w+\gamma\big)P_{pub} + P'_{pub}\Big)$$

$$T_1 \leftarrow xw\beta P \qquad T_2 \leftarrow x\delta P_{pub}$$

Outputs the offline ciphertext $\bar{\phi} = (u, x, \alpha, \beta, \gamma, \delta, U, R, T_0, T_1, T_2)$. Note that $e(wP + P_{pub}, P)$ can be pre-computed by the PKG as part of the *param* so that no pairing is needed in this phase.

Online-Encrypt: To encrypt a message $m \in \mathcal{M}$ to ID, at the online stage, computes:

$$t'_1 \leftarrow \beta^{-1}\Big(H_1(ID) - \alpha\Big) \bmod q \qquad t'_2 \leftarrow \delta^{-1}\Big(H_1(ID) - \gamma\Big) \bmod q$$

$$t \leftarrow H_2(m, R)x + u \bmod q \qquad c \leftarrow H_3(R) \oplus m$$

Outputs the ciphertext $\phi = (U, T_0, T_1, T_2, t'_1, t'_2, t, c)$.

Decrypt: To decrypt using secret key D_{ID}, computes

$$R \leftarrow e(T_0 + t'_1T_1 + t'_2T_2\ ,\ D_{ID}) \qquad m \leftarrow c \oplus H_3(R)$$

and checks whether

$$R^{H_2(m,R)} \stackrel{?}{=} e\big(tP + U\ ,\ wP + P_{pub}\big) \cdot e(P,P)^{-1} \tag{1}$$

Outputs m if it is equal. Otherwise outputs $\perp$.

Same as above, $e(P,P)$ can be pre-computed by the PKG as part of the *param*.

3.2 Security Analysis

Correctness. For the decrypt, we have

$$\begin{aligned}
& e(T_0 + t_1' T_1 + t_2' T_2, D_{ID}) \\
&= e\Bigg(x\Big(wH_1(ID)P + wP_{pub} + H_1(ID)P_{pub} + P'_{pub} \Big), \\
&\quad \Big(H_1(ID) + s \Big)^{-1} P \Bigg) \\
&= e\Bigg(x\Big(w\Big(H_1(ID) + s \Big)P + \Big(H_1(ID) + s \Big) P_{pub} \Big), \\
&\quad \Big(H_1(ID) + s \Big)^{-1} P \Bigg) \\
&= e\Bigg(x\Big(\Big(H_1(ID) + s \Big)\Big(wP + P_{pub} \Big) \Big), \Big(H_1(ID) + s \Big)^{-1} P \Bigg) \\
&= e\Bigg(x\Big(\Big(H_1(ID) + s \Big)\Big(w + s \Big) P \Big), \Big(H_1(ID) + s \Big)^{-1} P \Bigg) \\
&= e\Big((w+s)P, P \Big)^x \\
&= R
\end{aligned}$$

On the other side, let $h = H_2(m, R)$. We have

$$\begin{aligned}
& e\Big(tP + U, wP + P_{pub} \Big) \cdot e(P,P)^{-1} \\
&= e\Big(hxP + uP + \Big(w + s \Big)^{-1} P - uP, \Big(w + s \Big) P \Big) \cdot e(P,P)^{-1} \\
&= e\Big(hxP + \Big(w + s \Big)^{-1} P, \Big(w + s \Big) P \Big) \cdot e(P,P)^{-1} \\
&= e\Big(hxP, \Big(w + s \Big) P \Big) \cdot e\Big(\Big(w + s \Big)^{-1} P, \Big(w + s \Big) P \Big) \cdot e(P,P)^{-1} \\
&= \Big(e\Big(P, wP + P_{pub} \Big)^x \Big)^h = R^h
\end{aligned}$$

Theorem 1. *If there is a ID-IND-CCA adversary $\mathcal{A}$ of the proposed scheme that succeeds with probability ϵ, then there is a simulator $\mathcal{B}$ running in polynomial time that solves the $(\ell+1)$-BDHI problem with probability at least*

$$\epsilon \cdot \frac{1}{q_1} \Big(1 - \frac{q_d}{q} \Big)$$

where q_1, q_d are the number of queries allowed to the random oracle H_1 and decryption oracle respectively and we assume $q_1 = \ell$.

Proof. Setup: We follow the proof technique from [1]. Suppose $\mathcal{B}$ is given a random instance of the $(\ell+1)$-BDHI problem $(\hat{P}, \alpha\hat{P}, \alpha^2\hat{P}, \ldots, \alpha^{\ell}\hat{P}, \alpha^{\ell+1}\hat{P})$, $\mathcal{B}$ runs $\mathcal{A}$ as a subroutine to output $e(\hat{P},\hat{P})^{\frac{1}{\alpha}}$. $\mathcal{B}$ sets up a simulated environment for $\mathcal{A}$ as follow.

$\mathcal{B}$ first randomly selects $\pi \in_R \{1,\ldots,q_1\}$, $I_\pi \in_R \mathbb{Z}_q^*$ and $w_1,\ldots,w_{\pi-1}, w_{\pi+1},\ldots,w_\ell \in_R \mathbb{Z}_q^*$. For $i \in \{1,\ldots,\ell\}\setminus\{\pi\}$, it computes $I_i = I_\pi - w_i$. Construct a polynomial with degree $\ell-1$ as

$$f(z) = \prod_{i=1, i\neq\pi}^{\ell} (z + w_i)$$

to obtain $c_0,\ldots,c_{\ell-1} \in \mathbb{Z}_q^*$ such that $f(z) = \sum_{i=0}^{\ell-1} c_i z^i$. Then it sets generator $G = \sum_{i=0}^{\ell-1} c_i(\alpha^i\hat{P}) = f(\alpha)\hat{P}$.

For $i \in \{1,\ldots,\ell\}\setminus\{\pi\}$, $\mathcal{B}$ expands $f_i(z) = f(z)/(z+w_i) = \sum_{j=0}^{\ell-2} d_{i,j} z^j$ to obtain $d_{i,1},\ldots,d_{i,\ell-2} \in \mathbb{Z}_q^*$ and sets

$$\tilde{H}_i = \sum_{j=0}^{\ell-2} d_{i,j}(\alpha^j\hat{P}) = f_i(\alpha)\hat{P} = \frac{f(\alpha)}{\alpha+w_i}\hat{P} = \frac{1}{\alpha+w_i}G$$

It randomly chooses $\hat{w} \in \{1,\ldots,\ell\}\setminus\{\pi\}$, and computes the public key w, W, P_{pub} and P'_{pub} as

$$w = I_{\hat{w}} \qquad W = -\tilde{H}_{\hat{w}}$$

$$P_{pub} = -\alpha G - I_\pi G = (-\alpha - I_\pi)G \qquad P'_{pub} = \alpha^2 G + 2I_\pi\alpha G + I_\pi^2 G = (\alpha+I_\pi)^2 G$$

where $\alpha G = \sum_{i=0}^{\ell-1} c_i(\alpha^{i+1}\hat{P})$ and $\alpha^2 G = \sum_{i=0}^{\ell-1} c_i(\alpha^{i+2}\hat{P})$ so that its unknown master secret key msk is implicitly set to $x = -\alpha - I_\pi \in \mathbb{Z}_q^*$, while public parameter $param$ are set to $(G, P_{pub}, P'_{pub}, W, w)$ which are given to the adversary. For all $i \in \{1,\ldots,\ell\}\setminus\{\pi\}$, we have $(I_i, -\tilde{H}_i) = (I_i, \frac{1}{I_i+x}G)$.

Oracle Simulation: $\mathcal{B}$ first initializes a counter ν to 1 and starts $\mathcal{A}$. Throughout the game, we assume that H_1-queries are distinct, that the target identity ID^* is submitted to H_1 at some point.

1. *Random Oracle:* For H_1-queries (we denote ID_ν the input of the ν^{th} one of such queries), $\mathcal{B}$ answers I_ν and increments ν.
   For H_2-queries on input (m, R) and H_3-queries on input R, $\mathcal{B}$ returns the defined value if it exists and a randomly chosen $h_2 \in_R \mathbb{Z}_q^*$ for H_2 and $h_3 \in_R \{0,1\}^{n_m}$ for H_3 respectively, otherwise. $\mathcal{B}$ stores the information $(m, R, h_2, c = m \oplus h_3, \gamma = R^{h_2}\cdot e(G,G))$ in L_2 and (R, h_3) in L_3.
2. *Extraction Oracle:* On input ID_ν, if $\nu = \pi$, $\mathcal{B}$ aborts. Otherwise, it knows that $H_1(ID_\nu) = I_\nu$ and returns $-\tilde{H}_\nu = (1/(I_\nu + x))G$.
3. *Decryption Oracle:* On input a ciphertext $\phi = (U, T_0, T_1, T_2, t'_1, t'_2, t, c)$ for identity ID_ν, we assume that $\nu = \pi$ because otherwise $\mathcal{B}$ knows the receiver's

private key $D_{ID_\nu} = -\tilde{H}_\nu$ and can normally run the decryption algorithm. Let $\tilde{x} \in \mathbb{Z}_q$ such that

$$\begin{aligned} \tilde{x}W &= tG + U - W \\ \tilde{x}(w+x)^{-1}G &= tG + U - (w+x)^{-1}G \\ \tilde{x}G &= (w+x)(tG+U) - G \end{aligned} \tag{2}$$

Also let $T = T_0 + t'_1 T_1 + t'_2 T_2$, $G_{ID_\nu} = I_\nu G + P_{pub}$, and $h = H_2(m, R)$ (which is yet unknown to $\mathcal{B}$ at this moment). As all valid ciphertext satisfies

$$\begin{aligned} R^h &= e(tG+U, (w+x)G) \cdot e(G,G)^{-1} \\ e(hT, (I_\nu + x)^{-1}G) &= e((w+x)(tG+U), G) \cdot e(-G, G) \\ e((I_\nu + x)^{-1}hT, G) &= e((w+x)(tG+U) - G, G) \\ (I_\nu + s)^{-1}hT &= (w+x)(tG+U) - G \end{aligned} \tag{3}$$

Let $\tilde{x}' \in \mathbb{Z}_q$ such that

$$\begin{aligned} \tilde{x}' G_{ID_\nu} &= hT \\ \tilde{x}'(I_\nu + x)G &= hT \\ \tilde{x}'G &= (I_\nu + x)^{-1}hT \\ &= (w+x)(tG+U) - G \text{ (from equation(3))} \\ &= \tilde{x}G \text{ (from equation(2))} \\ \Rightarrow \tilde{x}' &= \tilde{x} \\ \Rightarrow \log_W(tG + U - W) &= \log_{G_{ID_\nu}}(hT) \end{aligned} \tag{4}$$

From equation (4), we have

$$e(hT, W) = e(G_{ID_\nu}, S - W) \tag{5}$$

where $S = tG + U$, which yields $e(hT, W) = e(G_{ID_\nu}, S) \cdot e(G_{ID_\nu}, W)^{-1}$.

The query is handled by computing $\gamma = e(S, wG + P_{pub})$, and search through the list L_2 for entries of the form $(m_i, R_i, h_{2,i}, c, \gamma)$ indexed by $i \in \{1, \ldots, q_2\}$. If none is found, ϕ is rejected. Otherwise, each one of them is further examined: for the corresponding indexes, $\mathcal{B}$ checks if

$$\frac{e(T, W)^{h_{2,i}}}{e(S, G_{ID_\nu})} = e(G_{ID_\nu}, W)^{-1} \tag{6}$$

meaning that equation (5) is satisfied. If the unique $i \in \{1, \ldots, q_2\}$ satisfying equation (6) is detected, the matching pair $(m_i, h_{2,i}, S)$ is returned. Otherwise ϕ is rejected.

Challenge: $\mathcal{A}$ outputs messages (m_0, m_1) and identities ID^* for which it never obtained ID^*'s private key. If $ID^* \neq ID_\pi$, $\mathcal{B}$ aborts. Otherwise it randomly

selects $t, t'_1, t'_2, \tilde{t}_0, \tilde{t}_1, \tilde{t}_2 \in_R \mathbb{Z}_q^*$, $c \in_R \{0,1\}^{n_m}$ and $U \in_R \mathbb{G}$. Computes $T_0 = \tilde{t}_0 G, T_1 = \tilde{t}_1 G, T_2 = \tilde{t}_2 G$ to return the challenge ciphertext $\phi^* = (U, t, T_0, T_1, T_2, t'_1, t'_2, c)$. Let $\xi = \tilde{t}_0 + t'_1\tilde{t}_1 + t'_2\tilde{t}_2$ and $T = -\xi G$. Since $x = -\alpha - I_\pi$, we let $\rho = \frac{\xi}{\alpha(w-\alpha-I_\pi)} = -\frac{\xi}{(I_\pi+x)(w+x)}$, we can check that

$$\begin{aligned} T = -\xi G &= -\alpha(w - \alpha - I_\pi)\rho G \\ &= (I_\pi + x)(w + x)\rho G \\ &= \rho(I_\pi w + (w + I_\pi)x + x^2)G \end{aligned}$$

$\mathcal{A}$ cannot recognize that ϕ^* is not a proper ciphertext unless it queries H_2 or H_3 on $e(wG + G_{pub}, G)^\rho$. Along the guess stage, its view is simulated as before and its output is ignored. Standard arguments can show that a successful $\mathcal{A}$ is very likely to query H_2 or H_3 on the input $e(G_{ID_\mu}, G)^\rho$ if the simulation is indistinguishable from a real attack environment.

Output Calculation: $\mathcal{B}$ fetches a random entry (m, R, h_2, c, γ) or $(R, \cdot)$ from the lists L_2 or L_3. With probability $1/(2q_2 + q_3)$, the chosen entry will contain the right element

$$R = e(wG + P_{pub}, G)^\rho = e(G, G)^{-\xi/(I_\pi+x)} = e(\hat{P}, \hat{P})^{f(\alpha)^2\xi/\alpha}$$

where $f(z) = \sum^* \ell - 1_{i=0} c_i z^i$ is the polynomial for which $G = f(\alpha)P$. The $(\ell+1)$-BDHI solution can be extracted by computing

$$\begin{aligned} &\left(\frac{R^{1/\xi}}{e\Big(\sum_{i=0}^{\ell-2} c_{i+1}(\alpha^i \hat{P}), c_0 \hat{P}\Big) e\Big(\sum_{j=0}^{\ell-2} c_{j+1}(\alpha^j)\hat{P}, G\Big)}\right)^{1/c_0^2} \\ &= \left(\frac{e(\hat{P}, \hat{P})^{f(\alpha)^2/\alpha}}{e(\hat{P}, \hat{P})^{c_0(c_1 + c_2\alpha + c_3\alpha^2 + \ldots c_{\ell-1}\alpha^{\ell-2})} e(\hat{P}, \hat{P})^{f(\alpha)(c_1 + c_2\alpha + c_3\alpha^2 + \ldots c_{\ell-1}\alpha^{\ell-2})}}\right)^{1/c_0^2} \\ &= \left(\frac{e(\hat{P}, \hat{P})^{f(\alpha)^2/\alpha}}{e(\hat{P}, \hat{P})^{\frac{c_0(c_1\alpha + c_2\alpha^2 + \ldots c_{\ell-1}\alpha^{\ell-1}) + f(\alpha)(c_1\alpha + c_2\alpha^2 + \ldots c_{\ell-1}\alpha^{\ell-1})}{\alpha}}}\right)^{1/c_0^2} \\ &= e(\hat{P}, \hat{P})^{\frac{f(\alpha)^2 - (c_1\alpha + c_2\alpha^2 + \ldots c_{\ell-1}\alpha^{\ell-1})(c_0 + f(\alpha))}{{c_0}^2\alpha}} \\ &= e(\hat{P}, \hat{P})^{\frac{{c_0}^2}{{c_0}^2\alpha}} \\ &= e(\hat{P}, \hat{P})^{1/\alpha} \end{aligned}$$

Probability Analysis: $\mathcal{B}$ only fails in providing a consistent simulation because one of the following independent events happen:

- E_1 : $\mathcal{A}$ does not choose to be challenged on ID_π.
- E_2 : A key extraction query is made on ID_π.
- E_3 : $\mathcal{B}$ rejects a valid ciphertext at some point of the game.

We have $\Pr[\neg E_1] = 1/q_1$ and $\neg E_1$ implies $\neg E_2$. Also observe that $\Pr[E_3] \leq q_d/q$. Combining together, the overall successful probability $\Pr[\neg E_1 \wedge \neg E_3]$ is at least

$$\frac{1}{q_1}\left(1 - \frac{q_d}{q}\right)$$

□

4 Comparison

There are only 2 existing online/offline IBE schemes, both of them are proposed by Guo, Mu and Chen in [6]. We use GMC-1 and GMC-2 to denote them. We also assume that $|\mathbb{G}| = 160$ bits, $|q| = 160$ bits, $|\mathbb{G}_T| = 1024$ bits and $|\mathcal{M}| = |q| = 160$ bits [1] for the following comparison. We denote by E the point multiplication in $\mathbb{G}$ or $\mathbb{G}_T$, ME the multi-point multiplication in $\mathbb{G}$ or $\mathbb{G}_T$ (which costs about 1.3 times more than a single point multiplication), M the point addition in $\mathbb{G}$ or $\mathbb{G}_T$ and m_c the modular computation in $\mathbb{Z}_q$.

Table 1. Comparison of computation cost and size

	GMC-1	GMC-2	Our scheme
Offline computation	$5E + 2ME$	$4E + 2ME$	$4E + 1ME$
Online computation	$1M + 2m_c$	$1M + 2m_c$	$3m_c$
Offline storage (bits)	2624	5056	2624
Ciphertext length (bits)	2144	6464	1280
Number of pairing for decryption	7	2	2
Security model	selective ID	standard	random oracle

We note that as GMC-1 requires an online/offline signature for encryption, we assume they use the most efficient one [3] which requires 320 bits for offline storage and 320 bits for signature length. The key generation and offline signing part require 1 E operation respectively. These costs have been added to the table.

From the above table, we can see that our scheme achieves the least computation and the smallest size in both offline and online stage, when compare to GMC-1 and GMC-2. There are a number of significant improvements:

1. First, we do not require any point addition operation (M operation) in the online encryption stage. Modular computation (m_c operation) is much faster than M operation. Other computations that required in our scheme such as hashing or XOR are negligible when compared to M operation. Thus our online encryption stage is much faster than GMC-1 and GMC-2.

[1] In our scheme, the message space can be any arbitrary length but the message space of GMC-1 and GMC-2 can be only set to the size of q, the order of group $\mathbb{G}_T$. In order to compare three schemes, we also set the message space of our scheme to 160 bits.

2. Second, the offline storage is as small as GMC-1 and about 50% smaller than GMC-2. This result is important in embedded device such as smart card or wireless sensor, where the storage is very limited.
3. Third, the ciphertext of our scheme is 40% smaller than GMC-1 and 80% smaller than GMC-2. This improvement is very significant in the environment where the communication bandwidth is very limited. On the other side, the number of pairing operations required in the decryption stage, is just 2, which is the same as GMC-2 while 3 times less than GMC-1. In other words, we combine and improve the efficiency advantages of both GMC-1 (short ciphertext) and GMC-2 (small number of pairing operations in decryption).
4. Forth, our scheme allows the message space to be any arbitrary length while the message space of GMC-1 and GMC-2 should be equal to the size of q, the order of group $\mathbb{G}_T$. Usually q is chosen as a 160-bits prime. That means the message space of GMC-1 and GMC-2 is 160 bits. If a larger message, say 1024 bits, is encrypted using GMC-1 or GMC-2, it must be divided into 7 parts ($\lceil \frac{1024}{160} \rceil = 7$) and carried out the encryption process 7 times. However in our scheme we just need to adjust the output length of the hash function H_3 to be 1024 and increase the size of the ciphertext from 1280 to 2144 bits ($1280 + (1024 - 160) = 2144$). Then we only need to execute the whole encryption process *once* (instead of 7 times, when compared to GMC-1 and GMC-2). Our scheme is particularly useful for a large message space.

We also remark that our scheme can be proven secure in the random oracle model, which is relatively stronger than the standard model or the selective-ID model. Although it is generally believed that random oracle model is not as secure as standard model theoretically, it still achieves an acceptable level of security. There are many applications that put efficiency as the most important factor. In these scenarios, schemes that are efficient but can be only proven secure in the random oracle model maybe a better choice.

5 Conclusion

In this paper we have proposed a new efficient identity-based online/offline encryption scheme. When compared to previous schemes, our scheme enjoys a number of significant improvements in efficiency. These improvements allow our scheme to be used in many practical scenarios such as smart card and wireless sensor networks. Our scheme can be proven secure in the random oracle model.

References

1. Barreto, P., Libert, B., McCullagh, N., Quisquater, J.: Efficient and provably-secure identity-based signature and signcryption from bilinear maps. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 515–532. Springer, Heidelberg (2005)

2. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
3. Boneh, D., Boyen, X.: Short signatures without random oracles the SDH assumption in bilinear groups. Journal of Cryptology 2, 149–177 (2008)
4. Chen, X., Zhang, F., Susilo, W., Mu, Y.: Efficient generic online/offline signatures without key exposure. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 18–30. Springer, Heidelberg (2007)
5. Even, S., Goldreich, O., Micali, S.: On-line/offline digital signatures. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 263–277. Springer, Heidelberg (1990)
6. Guo, F., Mu, Y., Chen, Z.: Identity-based online/offline encryption. In: Tsudik, G. (ed.) FC 2008. LNCS, vol. 5143, pp. 247–261. Springer, Heidelberg (2008)
7. Joye, M.: An efficient on-line/off-line signature scheme without random oracles. In: Franklin, M.K., Hui, L.C.K., Wong, D.S. (eds.) CANS 2008. LNCS, vol. 5339, pp. 98–107. Springer, Heidelberg (2008)
8. Kurosawa, K., Schmidt-Samoa, K.: New online/offline signature schemes without random oracles. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T.G. (eds.) PKC 2006. LNCS, vol. 3958, pp. 330–346. Springer, Heidelberg (2006)
9. Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)
10. Shamir, A., Tauman, Y.: Improved online/offline signature schemes. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 355–367. Springer, Heidelberg (2001)