An Efficient Signcryption Scheme from *q*-Diffie-Hellman Problems

Jayaprakash Kar

Information Security Research Group
Faculty of Computing & Information Technology
Department of Information Systems,
King Abdulaziz University, P.O.Box-80221, Jeddah-21589,
Kingdom of Saudi Arabia
{jayaprakashkar, jpkar.crypto}@yahoo.com

Abstract. Confidentiality and authenticity are two fundamental security requirement of Public key Cryptography. These are achieved by encryption scheme and digital signatures respectively. Here we present a provably secure signcryption scheme in random oracle model by modifying Libert et al's scheme [2]. Our scheme is more efficient and secure than Libert et al's scheme. Tan [1] proved that this scheme is not secure against non-adaptive chosen cipher text attacks. It has been also proved that the semantically secure symmetric encryption scheme proposed in the Libert et al's scheme is not sufficient to guarantee to be secure against adaptive chosen ciphertext attacks. Here we proposed a modified version of Libert et al's scheme. The security of which is proven using two assumptions, namely the Strong Diffie-Hellman (SDH) and Diffie-Hellman Inversion (DHI) in the random oracle model.

Keywords: Isomorphism, Bilinear Pairing, provably secure, DHP, SDH.

1 Introduction

Signcryption, first proposed by Zheng [3], is a cryptographic primitive that performs signature and encryption simultaneously, at lower computational costs and communication overheads than those required by the traditional sign-then-encrypt approach. Due to its advantages, there have been many signcryption schemes proposed after Zhengs publication. However, in some applications, sometimes people need both confidentiality and authentication and sometimes they just need confidentiality or authentication separately. For that case, applications must often contain at least three cryptographic primitives: signcryption, signature, and encryption, which will definitely increase the corresponding computation and implementation complexity and even will be infeasible in some resources-constrained environments such as embedded systems, sensor networks, and ubiquitous computing. Motivated by this, in 2006, Han et al. [4] proposed the concept of GSC which can implement the separate or joint encryption and signature functions in a single primitive.

2 Previous Works

In 1997, Zheng [3] introduced the concept of signcryption where both these properties are achieved in a single logical step, but in a more efficient way.

The notion of identity based cryptography was introduced by Shamir [5] in 1984. It is a form of public key cryptography in which the users do not obtain certificates for their public keys.

Instead, public keys are generated using arbitrary identifiers such as email addresses, telephone numbers and social security numbers that uniquely identifies a user in the system. This greatly reduces the problem of certificate management, considered to be cumbersome in PKI based systems. The private keys corresponding to the public keys are generated by a trusted authority called Private Key Generator (PKG). The first fully practical identity based encryption scheme was proposed by Boneh and Franklin [6] in 2001. Malone-Lee [7] proposed the first identity based signcryption scheme.

Yu et al. [9] proposed the first ID based signcryption scheme in the standard model and have been prove that the scheme is insecure against chosen message attack and adaptive cipher attack [8] [9] [10].

3 Preliminaries

3.1 Notation

Bilinear maps

Definition 1. *Let k be a security parameter and q be a k-bit prime number. Let $(\mathbb{G}_1, +)$ and $(\mathbb{G}_2, +)$ be two cyclic additive groups of same prime order p. $(\mathbb{G}_T, .)$ is an is a multiplicative group of prime order p and P, Q be generators of the group $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. We say $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ are* **Asymmetric Bilinear Map Groups** *if there exist a bilinear map $e : \mathbb{G}_1 X \mathbb{G}_2 \rightarrow \mathbb{G}_T$ satisfying the following properties:*

- **Bilinear***: $\forall (U, V) \in \mathbb{G}_1 X \mathbb{G}_2, \forall a, b \in \mathbb{Z}_p^*$, we have the relation $e(aU, bV) = e(U, V)^{ab}$.*
- **Non-degenerate***: $\forall U \in \mathbb{G}_1, e(U, V) = 1 \; \forall V \in \mathbb{G}_2 \Leftrightarrow U = 1_{\mathbb{G}_1}$.*
- **Computability***: $\forall (U, V) \in \mathbb{G}_1 X \mathbb{G}_2, e(U, V)$ is efficiently computable.*
- *There exists an efficient and publicly computable (but not necessarily invertible) isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ such that $\psi(Q) = P$.*

3.2 Mathematical Assumption

Definition 2. **q-Diffie-Hellman Inversion Problem(q-DHIP)***: Given a $(q+1)$-tuple of elements $(P_1, \alpha P_2, \alpha^2 P_2 \ldots \alpha^q P_2) \in \mathbb{G}_1 X \mathbb{G}_2^{q+1}$, computing $\frac{1}{\alpha} P_1 \in \mathbb{G}_1$ for a random $\alpha \in \mathbb{Z}_p^*$. We say that the (t, ϵ, q)-DHI assumption holds in $\mathbb{G}$ if no t-time algorithm has advantage at least ϵ in solving the q-DHI problem in $\mathbb{G}_1$.*

Definition 3. **q-Strong Diffie-Hellman Problem(q-SDHP)***: Let $(\mathbb{G}_1, \mathbb{G}_2)$ be bilinear map groups of generators P_1 and P_2. $\exists$ an isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ such that $\psi(P_1) = P_2$. Given a $(q+1)$-tuple $(P_1, \alpha P_2, \alpha^2 P_2 \ldots \alpha^q P_2) \in \mathbb{G}_1 X \mathbb{G}^{q+1}$, computing pair $(\frac{1}{\alpha + c} P, c) \in \mathbb{Z}_p^* X \mathbb{G}_1$ where $c \in \mathbb{Z}_p^*$. We say that the (t, ϵ, q)-SDH assumption holds in $\mathbb{G}_1$ if no t-time algorithm has advantage at least ϵ in solving the q-SDH problem in $\mathbb{G}_1$.*

Definition 4. **$(q+1)$-Exponent Problem(q-EP)***: Given a $(q+1)$-tuple $(P_1, \alpha P_2, \alpha^2 P_2 \ldots \alpha^q P_2) \in \mathbb{G}_1 X \mathbb{G}^{q+1}$, computing pair $\alpha^{q+1} P_1$ for a random $\alpha \in \mathbb{Z}_p^*$. We say that the $(t, \epsilon, q+1)$-EP assumption holds in $\mathbb{G}_1$ if no t-time algorithm has advantage at least ϵ in solving the $q+1$-exponent problem in $\mathbb{G}_1$.*

$(p+1)$-EP is no harder than the CDH problem and p-DHI problem is polynomially equivalent o $(p+1)$-EP.

4 Security Discussions

4.1 Security of Signcryption

Definition 5. (Confidentiality) *An signcryption scheme is semantically secure or has indistinguishbility against adaptive chosen ciphertext attack* ($IND-SC-CCA2$) *is no polynomially bounded(PPT) adversary has a non-negligible advantage in the following game.*

1. The challenger $\mathcal{C}$ runs the **Setup** algorithm with the security parameter k as input and generates a private and public key pair (sk_U, pk_U). sk_U is kept secret while pk_U is to be sends to the adversary $\mathcal{A}$.
2. The adversary $\mathcal{A}$ performs polynomial bounded number of queries to the oracles provided to $\mathcal{A}$ by $\mathcal{C}$. The description of the queries in the first phase are listed below:
   - **Signcryption oracle** : $\sigma \leftarrow Signcrypt(m, pk_i, sk_U)$. $\mathcal{A}$ submits a message $m \in \mathcal{M}$ and arbitrary public key pk_i to the challenger $\mathcal{C}$. $\mathcal{C}$ use the private key sk_U and runs the algorithm.
   - **DeSigncryption oracle**: $(m, \sigma) \leftarrow DeSigncrypt(\sigma, sk_U)$. $\mathcal{A}$ submits a ciphertext σ together with a sender's public key if the obtained signed plain-text is valid for recovered sender's and returns the symbol $\perp$ otherwise for rejection. $\mathcal{A}$ can present its queries adaptively *i.e* every request may depend on the response to the previous queries.
3. $\mathcal{A}$ chooses two messages $m_0, m_1 \in \mathcal{M}$ and an arbitrary private key sk_S of sender on which $\mathcal{A}$ wishes to be challenged. The challenger $\mathcal{C}$ flips a coins chooses a random bit $b \in \{0, 1\}$ and computes signcryption cipher-text $\sigma^* = Signcrypt(m_b, sk_S, pk_U)$ of m_b with the sender's private key sk_S under the attacked public key pk_U. The ciphertext σ is sent to the adversary $\mathcal{A}$ as a challenge.
4. $\mathcal{A}$ performs polynomial bounded number of new queries in the first stage, with the restrictions that $\mathcal{A}$ cannot query the de-signcryption oracle with (σ^*) and extraction oracle. These queries may be made adaptively *i.e* each query may depend on the answer to the previous queries.
5. At the end of the game $\mathcal{A}$ returns a bit $b^{'}$ and wins the game if $b^{'} = b$. The success probability is defined by:

$$Adv^{IND-SC-CCA2}(\mathcal{A}) = |Pr[b^{'} = b] - \tfrac{1}{2}|$$

   Here Adv is called the advantage for the adversary in the above game.

Definition 6. (Unforgeability) *An signcryption scheme (SC) is said to be existentially unforgeable against adaptive chosen-messages attacks (EUF-SC-CMA) if no polynomial bounded adversary (PPT) has a non-negligible advantage in the following game:*

1. The challenger $\mathcal{C}$ generates a private and public key pair (sk_U, pk_U). sk_U is kept secret while pk_U is to be sends to the adversary $\mathcal{A}$.
2. $\mathcal{A}$ performs polynomial bounded number of queries to the following oracles which are simulated by the challenger $\mathcal{C}$. The queries may be adaptive *i.e* the current query may depend on the previous query responses.
3. Challenger $\mathcal{C}$ generates a key pair (sk_U, pk_U) and pk_U is given to the forger $\mathcal{F}$.
4. $\mathcal{F}$ adaptively performs queries to the same oracles as in the above definition.
5. $\mathcal{F}$ eventually produces a ciphertext σ and a key pair (sk_R, pk_R) and wins if the result of $Designcrypt(\sigma, sk_R)$ is a triple (m, s, pk_U) such that the pair (m, s) is valid for the public key pk_U and no query to the signcryption oracle involving the message m and some receiver's public key pk_R^* return in a ciphertext σ^* for which the output of $DeSincrypt(\sigma^*, sk_R^*)$ is (m, s, pk_U).

5 Previous Work

6 Libert et al.'s Scheme

In this section we have described the signcrytion scheme proposed by Benoit Libert and jean-Jacques Quisquater [2]. This scheme consists of following four PPT algorithms

- **Common-Keygen**: Given a security parameter k, this algorithm outputs a k-bit prime number p and the description of bilinear map groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ of order p. Let l_1 and l_2 be polynomials in k respectively denoting the bit-length of elements from $\mathbb{G}_1$ and $\mathbb{G}_2$. The algorithm also chooses generators $P_1 \in \mathbb{G}_1$ and P_2 with $P_1 = \psi(P_2)$, hash functions $\mathcal{H}_1 : \{0,1\} \rightarrow \mathbb{Z}_p, \mathcal{H}_2 : \mathbb{G}^3 \rightarrow \{0,1\}^{k+l}$ and $\mathcal{H}_3 : \{0,1\}^k \rightarrow \{0,1\}^\lambda$. Let $\mathcal{H}^{'}$ be a pseudo-random function, $\mathcal{H}^{'} : \{0,1\}^* \rightarrow \{0,1\}$. The common public parameters are $param = \{p, \mathbb{G}_1, \mathbb{G}_2, P_1, P_2, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}^{'}, n\}$. Where n denotes the length of message $m \in \mathcal{M}$ *i.e* $|m| = n$.
- **Keygen** each users selects $x_u \in \mathbb{Z}_p$ and computes $y_u = x_u \cdot P_2 \in \mathbb{G}_2$ obtains a public and private keys pairs $(pk_u, sk_u) = (Y_u, x_u)$.
- **Signcrypt** Given a message $m \in \{0,1\}^n$, the recipient's public key Y_R and her private key x_s, the sender does the following:
  1. Select $\gamma \in \mathbb{Z}_p^*$ and compute $r = \gamma\mathcal{H}_1((b_m \| m \| Y_S) + x_s)^{-1} \bmod p$, where $b_m = \mathcal{H}^{'}(x_s, m) \in \{0,1\}$.
  2. Set $C_1 = rP_1 \in \mathbb{G}_1, C_2 = (\gamma \| b_m) \oplus \mathcal{H}_2(C_1 \| Y_R r\psi(Y_R)) \in \{0,1\}^{k+l}$ and $C_3 = (m \| Y_S) \cdot \mathcal{H}_3(\gamma) \in \{0,1\}^{n+l_2}$. The ciphertext is $C = \langle C_1, C_2, C_3 \rangle$.
- **DeSigncrypt** Given the input ciphertext $C = \langle C_1, C_2, C_3 \rangle$
  1. Compute $(\gamma \| b_m) = C_2 \oplus \mathcal{H}_2(C_1 \| Y_R \| x_R C_1) \in \{0,1\}^k$.
  2. Compute $(m \| Y_S) = C_3 \oplus \mathcal{H}_3(\gamma) \in \{0,1\}^{n+l_2}$.
  3. Compute $\sigma = \gamma^{-1} C_1 \in \mathbb{G}_1$ and accept the message if $e(\sigma, Y_S + \mathcal{H}(b_m \| m \| Y_S) \cdot P_2) = e(P_1, P_2)$.

7 Security Analysis of Libert et al.'s Scheme

Chilk-How TAN [1] proves that Libert et al's scheme is not secure against non-adaptive chosen ciphertext and is not sufficiently guarantee their signcryption to be secure against adaptive chosen ciphertext attacks. The proof are shown by the following two claims.

Claim 1: The Libert-Quisquaters q-DH signcryption scheme is not secure against non-adaptive chosen ciphertext attacks.

Assume that given the receiver's public key x_r and the adversary e$\mathcal{A}$ first chooses a sender secret key x_s and two equal length messages m_0 and m_1 such that $b_{m_0} = \mathcal{H}^{'}(x_s, m_0) = 0$ and $b_{m_1} = \mathcal{H}^{'}(x_s, m_1) = 1$ and send x_s, m_0 and m_1 to the challenger. The challenger then compute the challenge ciphertext $C^* = \langle b^*_{m_b}, C^*_1, C^*_2, C^*_3 \rangle$ where $b \in \{0,1\}$. Upon receipt of the challenge ciphertext C^*, then b_{m_b} must be equal to either b_{m_0} or b_{m_1}. Hence the adversary $\mathcal{A}$ can make a correct guess $b^{'}$ which is equal to b. Therefore, we conclude that the Libert-Quisquater q-DH signcryption scheme is not secure against non-adaptive chosen cipher-text attacks.

Claim-2 :The semantically secure symmetric encryption scheme $(\mathcal{E}, \mathcal{D})$ in the Libert-Quisquaters q-DH signcryption scheme does not sufficiently guarantee their signcryption to be secure against adaptive chosen ciphertext attacks.

Proof: Let the semantically secure symmetric encryption scheme be $(\bar{\mathcal{E}}, \bar{\mathcal{D}})$ is constructed. Let

the receiver's public key X_r, the adversary $\mathcal{A}$ first selects a sender's secret key x_S and two equal length messages m_0 and m_1 and send to the challenger $\mathcal{C}$. The challenger then computes the challenge ciphertext $\bar{C} = \langle \bar{b_{m_b}}, \bar{C_1}, \bar{C_2}, \bar{C_3} \rangle$, where $b \in \{0,1\}$. Upon receipt of the challenge cipher $\bar{C} = \langle \bar{b_{m_b}}, \bar{C_1}, \bar{C_2}, \bar{C_3} \rangle$, the adversary first make a wild guess that b to be 0 and construct a new cipher text by choosing a random message $\hat{m}$ whose length is equal to that of m_0 and computing the following:

$$\hat{X_S} = X_s g^{\mathcal{H}_1(\bar{b_{m_b}} \| m_0) - \mathcal{H}_1(0 \| \hat{m})}$$

$$\hat{C_3} = \bar{C_3} \oplus (m_0 \oplus \hat{m}) \| (X_S \oplus \hat{X_S})$$

Then the adversary $\mathcal{A}$ sent the ciphertext $\hat{C} = \langle 0, \bar{C_1}, \bar{C_2}, \hat{C_3} \rangle$ to the challenger for de-cryption. Upon receipt the query, the challenger computes $\hat{w} = \bar{C_2} \oplus \mathcal{H}_2(\bar{C_1}, X_R, \bar{C_1}^{x_R})$. If $\hat{w} \notin \mathbb{Z}_p^*$, then returns $\perp$, otherwise computes the following:

$$k = \mathcal{H}_3(\hat{w}), m^{'} \| X_S^{'} = \bar{\mathcal{D}}_k(\hat{C_3}), \hat{\sigma} = \bar{C_1}^{\hat{w}-1}.$$

If $e(\hat{\sigma}, X_S^{'} g^{\mathcal{H}_1(0 \| m^{'})}) = e(g,g)$, then the challenger returns the message $m^{'}$, otherwise reject the message. If the response from the challenger is $m^{'}$ which is equal to $\hat{m}$, then adversary $\mathcal{A}$ will know that m_0 is the plaintext for the challenger ciphertext . If the response is rejected, then m_1 is the plaintext for the challenge ciphertext. Hence the adversary can make a correct guess of b.

8 Signcryption Scheme

8.1 Framework of Signcryption Scheme

An signcryption scheme comprises following four probabilistic polynomial time (PPT) algorithms:

- **Setup**: $(p) \leftarrow \text{Set}(1^k)$ takes a security parameter $k \in \mathbb{N}$ and generates a k-bit prime number p. Given a security parameter k, this algorithm outputs a k-bit prime number p.
- **Keygen**: $(Q_u, x_u) \leftarrow KeyGen(1^k, param)$ takes a security parameter k, the global parameters $param$, each user select $x_u \in \mathbb{Z}_p^*$ randomly, generates the public and private key pairs (Q_u, x_u) by computing $Q_u = x_u P_2 \in \mathbb{G}_2$.
- **Signcrypt**: $C \leftarrow Signcrypt(1^k, param, Q_R, x_S, m)$. Given a message $m \in \{0,1\}^n$, where $m \in \mathcal{M}$, the algorithm takes a security parameter k, global parameters $param$, recipient's public key Q_R and sender's private key x_S generate ciphertext C.
- **De–Signcrypt**: $(m, \sigma)/\perp \leftarrow DeSigncrypt(1^k, param, C)$ takes a security parameter k, the global parameters $param$, cipher text C, private key of the receiver x_R to generate the plain-text m and a signature σ or $\perp$ otherwise.
- **Verify**: $(Valid/\perp) \leftarrow Verify(1^k, param, m, \sigma)$. The algorithm takes a security parameter k, a global parameters $param$, message m, signature σ outputs $Valid$ or $\perp$ for invalid signature.

9 Proposed Signcryption Scheme

The scheme comprise four randomized polynomials algorithms.

- **Setup**: $(p) \leftarrow \text{Set}(1^k)$ takes a security parameter $k \in \mathbb{N}$ and generates a k-bit prime number p. Given a security parameter k, this algorithm outputs a k-bit prime number p . The algorithm chooses groups $(\mathbb{G}_1, \mathbb{G}_2)$ of prime order p. A bilinear map $\hat{e} : \mathbb{G}_1 X \mathbb{G}_1 \rightarrow \mathbb{G}_2$. Let l_1 and l_2 be polynomials in k respectively denoting the bit-length of elements from $\mathbb{G}_1$ and $\mathbb{G}_2$. The algorithm also chooses generators $P_1 \in \mathbb{G}_1$ and P_2 with $P_1 = \psi(P_2)$, collision resistant hash functions $\mathcal{H}_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*, \mathcal{H}_2 : \mathbb{G}_1 X \mathbb{G}_1 X \mathbb{G}_1 \rightarrow \{0,1\}^{k+l}$, $\mathcal{H}_3 : \{0,1\}^k \rightarrow \{0,1\}^\mu$ and $\mathcal{H}^{'} : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$. The common public parameters are

$$param = \{p, \mathbb{G}_1, \mathbb{G}_2, P_1, P_2, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}^{'}, n\}.$$

Where n denotes the length of message $m \in \mathcal{M}$ *i.e* $|m| = n$.

- **Keygen** : Each user select $x_u \in \mathbb{Z}_p^*$ randomly and computes $Q_u = x_u P_2$ generates the public and private key pairs (Q_u, x_u).
- **Signcryption**: Let the given message $m \in \mathcal{M}$, The sender performs the following steps.
  1. Select $\lambda \in \mathbb{Z}_p^*$ randomly and computes the following
  $$U = x_s P_1 \text{ and } h^{'} = m \| \mathcal{H}^{'}(U)$$
  2. $\xi = \lambda(\mathcal{H}_1(h^{'} \| Q_s) + x_s)^{-1} \bmod p$
  3. $C_1 = \xi P_1 \in \{0,1\}^{l_1}$, $C_2 = (\lambda \| h^{'}) \oplus \mathcal{H}_2(C_1 \| Q_R \| \xi\psi(Q_R)) \in \{0,1\}^{k+l}$ and $C_3 = (m \| Q_S) \oplus \mathcal{H}_3(\lambda) \in \{0,1\}^{n+l_2}$.
  The ciphertext is $C = \langle C_1, C_2, C_3 \rangle$
- **De–Signcryption**:Given $C = \langle C_1, C_2, C_3 \rangle$, compute the following
  1. $(\lambda \| h') = C_2 \oplus \mathcal{H}_2(C_1 \| Q_R \| x_r C_1) \in \{0,1\}^k$. Return $\perp$ if $\lambda \notin \mathbb{Z}_p^*$.
  2. $(m \| Q_s) = C_3 \oplus \mathcal{H}_3(\lambda) \in \{0,1\}^{n+l_2}$.
  3. Computes $\sigma = \lambda^{-1} C_1 \in \mathbb{G}_1$. Accept the message if the following equation holds

$$e(\sigma, Q_S + \mathcal{H}_1(h' \| Q_S) P_2) = e(P_1, P_2) \tag{1}$$

10 Proof of Correctness

$$e(\sigma, Q_S + \mathcal{H}_1(h^{'} \| Q_S) P_2) = e(P_1, P_2)$$

$$\begin{aligned}
& e(\sigma, Q_S + \mathcal{H}_1(h^{'} \| Q_S) P_2) \\
&= e(\lambda^{-1} C_1, Q_S + \mathcal{H}_1(C_2 \| Q_S) P_2) \\
&= e((\xi \mathcal{H}_1(C_1 \| Q_S) + x_S)^{-1} (\xi P_1), x_S P_2 + \mathcal{H}_1(h' \| Q_S) P_2) \\
&= e((\xi^{-1}(\mathcal{H}_1(h^{'} \| Q_S) + x_S)^{-1} (\xi P_1), x_S P_2 + \mathcal{H}_1(h^{'} \| Q_S) P_2) \\
&= e((\mathcal{H}_1(h^{'} \| Q_S) + x_S)^{-1} P_1, x_S P_2 + \mathcal{H}_1(h^{'} \| Q_S) P_2) \\
&= e((\mathcal{H}_1(C_1 \| Q_S) + x_S)^{-1} P_1, (x_S + \mathcal{H}_1(C_1 \| Q_S)) P_2) \\
&= e(P_1, P_2)
\end{aligned}$$

10.1 Security Analysis

Theorem 1 *Assume that an adversary $\mathcal{A}$ has non-negligible advantage ϵ. in breaking the IND-SC-CCA2 security of the scheme when running in time T asking $q_{\mathcal{H}_i}$ queries to random oracles $\mathcal{H}_i$ for $i = 1, 2, 3$ and $q_{\mathcal{H}'}$ queries to random oracle $\mathcal{H}'$. q_{se} sincryption (signature/encryption) queries and q_{dv} desincryption (decryption/verification) queries. Then there exists a PPT algorithm $\mathcal{B}$ to solve the q-Diffie-Hellman Inversion problem (qDHI) for $q = q_{\mathcal{H}_1}$ with advantage*

$$\epsilon^{'} \geq \epsilon - \frac{q_{dv}}{2^k} - \frac{q_{\mathcal{H}_3}}{2^{n+l_2}}$$

when running in time $T^{'} \leq +O(q_{\mathcal{H}_1}^2 T_m) + 2q_{\mathcal{H}_2} T_p$, where T_m is the maximal cost of scalar multiplication in $\mathbb{G}_1$ and $\mathbb{G}_2$, T_p being the time for bilinear map evaluation.

Proof:

- **Setup**: Suppose $\mathcal{B}$ is given a random instances of the $(q+1)$-DHI problem $(P, Q, \alpha Q, \alpha^2 Q \ldots \alpha^q Q) \in (\mathbb{G}_1, \mathbb{G}_2)$ which is equivalent to q-DHI problem to compute $\alpha^{q+1} P \in \mathbb{G}_1$. $\mathcal{B}$ runs $\mathcal{A}$ as a subroutine to solve the above random instances and act as $\mathcal{A}$'s challenger in the IND-SC-CCA2 game.

Initial: In a preparation phase $\mathcal{B}$ chooses $l \in \{1, 2 \dots q_{\mathcal{H}_i}\}$, elements $e_l \in \mathbb{Z}_p^*$ and $w_1, w_2 \dots w_{l-1}, w_{l+1} \dots w_q \in \mathbb{Z}_q^*$ randomly. For $i = 1, 2 \dots l-1, l+1 \dots q$, it computes $e_i = e_l - w_i$. $\mathcal{B}$ uses its input to compute a generator $M \in \mathbb{G}_2$ and $N = \psi(M) \in \mathbb{G}_1$ together with public key $R = \alpha M \in \mathbb{G}_2$ such that it knows $q-1$ pairs $(w_i, V_i = \frac{1}{\alpha + w_i} M)$ for $i \in \{1, 2 \dots q\}$ l. We can apply the proof technique of Boench and Boyen 2004, $\mathcal{B}$ expands the polynomials as

$$f(z) = \prod_{i=1, i \neq l}^{q} (z + w_i) = \sum_{j=0}^{q-1} c_j z^j$$

To obtain $c_0, c_1 \dots c_{q-1} \in \mathbb{Z}_p^*$ such that $f(z) = \sum_{i=0}^{q-1} (c_i z^i)$. Then it sets generator $M \in \mathbb{G}_2$ and the public key R are then obtains as

$$M = \sum_{i=0}^{q-1} c_i (\alpha^i Q) = f(\alpha) Q \text{ and } R = \sum_{i=1}^{q} c_{i-1} (\alpha^i Q) = \alpha f(\alpha) Q = \alpha M$$

By applying similar technique of proof in [2], we can obtain the pairs $(w_i, V_i = \frac{1}{w_i + \alpha} M)$ by expanding $f_i(z) = \frac{f(z)}{z + w_i} = \sum_{i=0}^{q-1} d_i z^i$ for $i \in \{1, 2 \dots q\}$ q and computing as

$$V_i = \sum_{j=0}^{q-2} d_j (\alpha^j Q) = f_j(\alpha) Q = \frac{f(\alpha)}{\alpha + w_i} Q = \frac{1}{\alpha + w_i} M.$$

The adversary $\mathcal{A}$ is then initialized with the generator $M \in \mathbb{G}_2$ and $N = \psi(M) \in \mathbb{G}_1$ and on the public key $R \in \mathbb{G}_2$. She will be given access to the oracles. The oracles are simulated by $\mathcal{B}$ and maintain a lists L_1 and L_2 that are initially empty. These are used to keep track of answers to queries asked by $\mathcal{A}$ to oracles $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$.

Oracle Simulation:

1. $\mathcal{H}^{'}$**-Queries**: For $\mathcal{H}^{'}$-queries on input $U \in \mathbb{G}_1$, $\mathcal{B}$ first checks if $R = x_i M$. Here it success and $\mathcal{B}$ can easily compute $\alpha^{q+1} P$.
2. $\mathcal{H}_1$**-Queries**: These queries are indexed by counter t that is initially set to 1. When a $(d \| R)$ is submitted in $\mathcal{H}_1$ query for the first time, $\mathcal{B}$ checks whether d equal to the string $h^{'}$. If $d = h^{'}$, $\mathcal{B}$ returns w_t and increments t (in such a way that $\mathcal{B}$ is able to generate a valid signature on m. Otherwise, $\mathcal{B}$ returns a random element $c \in \mathbb{Z}_p^*$ stores (d, R, c) in L_1.
3. $\mathcal{H}_2$**-Queries**: On input $Q_{1,i} \| Q_{2,i} \| Q_{3,i} \in \mathbb{G}_1 X \mathbb{G}_2 X \mathbb{G}_1$, $\mathcal{B}$ checks if 4-uple $(M, Q_{1,i}, Q_{2,i}, Q_{3,i})$ is a valid co-Diffie Hellman tuples, we can write $Q_{3,i} = co{-}DH_M(Q_{1,i}, Q_{2,i})$ by verifying the following equation.
$$e(Q_{1,i}, Q_{2,i}) = e(Q_{3,i}, M) \tag{2}$$
If the verification is correct, $\mathcal{B}$ checks if L_2 contains a record $(Q_{1,i}, Q_{2,i}, Q_{3,i}, \beta_i)$ is in L_2, $\mathcal{B}$ returns a strings $\beta_i \in \{0,1\}^{k+1}$ and insert $(Q_{1,i}, Q_{2,i}, Q_{3,i}, \beta_i, 1)$ in L_2. If $(M, Q_{1,i}, Q_{2,i}, Q_{3,i})$ is not a co-DH tuple the entry $(Q_{1,i}, Q_{2,i}, Q_{3,i}, \beta_i, 0)$ is added in L_2. At most $2q_{\mathcal{H}_2}$ pairings must be computed overall.
4. $\mathcal{H}_3$**-Queries**: On input the random element $\lambda \in \{0,1\}^{n+l_2}$ return the hashed valued.
5. **Signcryption Oracle**: Signcryption queries on a plaintext m, for an arbitrary receiver's key Q: we assume that m was previously submitted in a $\mathcal{H}_1$ query and that the message dependent hashed value was previously defined. Since $\mathcal{H}_1(h^{'} \| R)$ was or will be defined to be w_j for some $j \in \{1, 2 \dots t\}$. $\mathcal{B}$ knows that previously computed $N_j = (\frac{1}{w_j + \alpha}) N$ appears as a valid signature on m from the adversary $\mathcal{A}$'s views. So it computes $C_1 = \lambda N_j \in \mathbb{G}_1$ for some $\lambda \in \mathbb{Z}_p^*$, obtains $k = \mathcal{H}_3(\lambda) \in \{0,1\}^{n+l_2}$ through $\mathcal{H}_3$ simulation and computes $C_3 = (m \| R) \oplus k \in \{0,1\}^{n+l_2}$. It then checks if L_1 contains a record $(C_1, Q, Q_3, \beta, 1)$ indicating that $Q_3 = \text{co}{-}\text{DH}_M(C_1, Q)$. If this entry exits, $\mathcal{B}$ returns $C = \langle C_1, C_2, C_3 \rangle$ with $C_2 = (\lambda \| h^{'}) \oplus \beta \in \{0,1\}^{k+1}$. Otherwise returns $C = \langle C_1, C_2, C_3 \rangle$ for a random $C_2 = \in \{0,1\}^{k+1}$ and inserts $(C_1, Q \dots (\lambda \| h^{'} \oplus C_2))$ in the special list L_2.

6. **De-signcryption Oracle**:When $\mathcal{A}$ submits a ciphertext $C = \langle C_1, C_2, C_3\rangle$, $\mathcal{B}$ checks whether L_1 contains the unique entry $(C_1, R, Q, \beta, 1)$ for some $Q \in \mathbb{G}_1$ and $\beta \in \{0,1\}^{k+1}$ indicating that $Q = \text{co}-\text{DH}_M$:
   (a) If it does, $\mathcal{B}$ obtains $(\lambda \| h^{'}) = C_2 \oplus \beta \in \{0,1\}^{n+l_2}$. C is also rejected if R_S is not a $\mathbb{G}_2$. Finally, $\mathcal{B}$ extracts $\sigma = \lambda^{-1} C_1$ and returns the plaintext $m \in \{0,1\}^n$ and signature σ together with the sender's public key $R_S \in \mathbb{G}_2$ if the verification equation-1 holds.
   (b) If it does not, $\mathcal{B}$ picks a random β picks a random $\beta \in \{0,1\}^{k+1}$ inserts $(C_1, R \ldots \beta)$ into the list L_2 so that a subsequent $\mathcal{H}_2$-query on $(C_1, R$,co$-DH_M(C_1, R))$ will receive β as an answer, before finalizing the job with random element β. It checks $(\lambda \| h^{'}) = C_2 \oplus \beta \in \{0,1\}^{k+1}$, $k = \mathcal{H}_3(\lambda)$ and so on. The extracted signature $\sigma = \lambda^{-1} C_1$ is checked above.

Probability Analysis: $\mathcal{A}$ returns messages m_0, m_1 and a sender's private key $x_s \in \mathbb{Z}_p^*$. At this moment $\mathcal{B}$ chooses a random $\theta \in \mathbb{Z}_p^*$ and computes $C_1^* = (x + \theta) \cdot N \in \mathbb{G}_1$ as $C_1^* = \psi(R) + \theta \cdot N$. It expands the polynomials $f^{'}(z) = f(z)(z+\theta) = \sum_{j=0}^{q} f_j z^j$ and returns the challenge $C^* = \langle C_1^*, C_2^*, C_3^*\rangle$, where $b^* \in \{0,1\}, C_2^* \in \{0,1\}^{k+1}, C_3^* = (m_b \| x_s M) \oplus k$ for a random bit $b \in \{0,1\}$ and $k \in \{0,1\}^{n+l_2}$. Its probability is $q_{dv}/2^k$. where C^* is submitted in designcryption query before the challenge phase, $\mathcal{B}$ aborts. The probability for $\mathcal{H}_3$-query is atmost $q_{\mathcal{H}_3}/2^{n+l_2}$. Therefore we have $\epsilon^{'} \geq \epsilon - \frac{q_{dv}}{2^k} - \frac{q_{\mathcal{H}_3}}{2^{n+l_2}}$.

The bound on $\mathcal{B}$'s computation time derives from the fact that $\mathcal{B}$ needs q and two scalar multiplications with q elements in $\mathbb{G}_1$ and $\mathbb{G}_2$. Total cost is $\mathcal{O}(q_{\mathcal{H}_1}^2)$ scalar multiplication in $\mathbb{G}_1$ and $\mathbb{G}_2$. For $\mathcal{H}_2$ queries, $\mathcal{B}$ needs a cost $\mathcal{O}(q_{\mathcal{H}_2}^2)$ pairings.

Theorem 2 *Assume that a forger $\mathcal{F}$ has non-negligible advantage ϵ. in breaking the EUF-SC-CMA security of the scheme when running in time T asking $q_{\mathcal{H}_i}$ queries to random oracles $\mathcal{H}_i$ for $i = 1, 2, 3$ and and $q_{\mathcal{H}'}$ queries to random oracle $\mathcal{H}'$. Let q_{se} be the signature/encryption queries and q_{dv} decryption/verification queries. Then there exists a PPT algorithm $\mathcal{B}$ to solve the q-Diffie-Hellman Inversion problem (qDHI) for $q = q_{\mathcal{H}_1}$ with advantage*

$$\epsilon^{'} \geq \epsilon - \frac{1}{2^k} - \frac{1}{2^{n+l_2}}$$

when running in time $T^{'} \leq +O(q_{\mathcal{H}_1}^2 T_m) + 2q_{\mathcal{H}_2} T_p$, where T_m is the maximal cost of scalar multiplication in $\mathbb{G}_1$ and $\mathbb{G}_2$, T_p being the time for bilinear map evaluation.

Proof: We shows $\mathcal{B}$ can provide a faithful simulation $\mathcal{F}$ and solve the $(q+1)$-DHI problem by interacting with $\mathcal{F}$. $\mathcal{B}$ takes as input random instances $(P, Q, \alpha Q, \alpha^2 Q \ldots \alpha^q Q) \in (\mathbb{G}_1, \mathbb{G}_2)$ which is equivalent to q-DHI problem to compute $\alpha^{q+1} P \in \mathbb{G}_1$. $\mathcal{B}$ runs $\mathcal{A}$ as a subroutine to solve the above random instances and act as $\mathcal{F}$'s challenger in the EUF-SC-CMA game.

Eventually, $\mathcal{F}$ halts and outputs a forged signature embedded into a cipher text $C^* = \langle C_1^*, C_2^* C_3^*\rangle$ and an arbitrary recipient's key pairs (x_R^*, Q_R^*), where $Q_R^* = x_R^* M$. These allows $\mathcal{B}$ recover the fake pairs (m^*, σ^*), where $\sigma^* = (\mathcal{H}_1(h^{'} \| R) + x)^{-1} L$ embedded into C^*, $\mathcal{B}$ can extract a solution to q-SDH as follows: if $\mathcal{F}$ is successful, $\mathcal{B}$ recovers a valid message-signature pair for the sender's public key R by computing $\lambda^* \| h^{'} = C_2 \oplus \mathcal{H}_2(C_1^* \| Q_R^* \| x_R^* C_1)$, $m^* \| R = C_3^*(\lambda^*)$ and $\sigma^* = \lambda^{*-1} C_1^*$. A q-SDH pair $\langle \mathcal{H}_1^*, L^*\rangle$ can be extracted by expanding $\frac{f(z)}{z + \mathcal{H}_1^*}$ into $\mathcal{H}^* = \mathcal{H}_1(h^{*\prime}, Q_S^*)$ and computing $N^* = \frac{1}{\lambda - 1}[\sigma^* - \sum_{i=0}^{q-1} \lambda_i \psi(\alpha_i Q)]$.

11 Conclusion

In this article we have proposed an efficient signcryption scheme which can be implemented on low processor and power-constrained mobile devices. The scheme is secure against existential forgery under chosen message attacks and adaptively chosen ciphertext attacks under the notions of indistingushability of ciphertext in the random oracle model.

References

1. Chik-How-TAN Security Analysis of Signcryption Scheme from q-Diffie Hellman problems, Special Section on Cryptography and Information Security, IEICE TRANS.Fundamentals, Vol-89 No-1 Jan 2006.
2. Benoit libert and J Quisquater Improved Signcryption from q-Diffie-hellman Problems Proceeding SCN 04, 2004.
3. Y. Zheng Digital signcryption or how to achieve cost (signature & encryption)¡¡cost(signature) + cost (encryption). In Advances in Cryptology CRYPTO1997, Lecture Notes in Computer Science, vol. 1294. Springer: Heidelberg, 1997; 165-179.
4. Y. Han and X. Yang ECGSC: Elliptic curve based generalized signcryption scheme. Cryptology ePrint Archive, Report 2006/126, 2006, http:/eprint.iacr.org.
5. Adi Shamir. Identity-based cryptosystems and signature schemes. In CRYPTO, pages 4753, 1984.
6. Dan Boneh and Matthew K. Franklin Identity-based encryption from the weil pairing. In CRYPTO, volume 2139 of Lecture Notes in Computer Science, pages 213229. Springer, 2001.
7. John Malone-lee Identity-based signcryption. In In Proceedings of Public Key Cryptography - PKC 2005, LNCS 3386, pages 362379. Springer, 2002.
8. Ren Yanli and Gu Dawu Efficient identity based signature/signcryption scheme in the standard model. In The First International Symposium on Data, Privacy, and E-Commerce, 2007, ISDPE 2007, pages 133-137, 2007.
9. Yong Yu, Bo Yang, Ying Sun, and Shenglin Zhu Identity based signcryption scheme without random oracles, Computer Standards & Interfaces, 31(1):56-62, 2009.
10. Xing Wang and Hai feng Qian Attacks against two identity-based signcryption schemes. In Second International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC), volume 1, pages 24-27, april 2010.