Reducing Security Overhead for Mobile Networks *

Fangguo Zhang[1], Yi Mu[2], and Willy Susilo[2]
[1] Department of Electronics and Communication Engineering
Sun Yat-Sen University, Guangzhou 510275, P.R. China
isdzhfg@zsu.edu.cn
[2] School of IT and Computer Science
University of Wollongong, Wollongong, NSW 2522, Australia
{ymu,wsusilo}@uow.edu.au

Abstract

Security of mobile communications comes with the cost of computational overhead. Reducing the overhead in security computations is critical to ensure the overall performance of a mobile network. In this paper, we present the notion of online/offline signcryption, where most of computations are carried out offline and the online part of our scheme does not require any exponent computations and therefore is very efficient. Our scheme allows any third party to verify the encryption without compromising confidentiality. We also show that our scheme is secure against existential forgery under chosen message attacks and adaptively chosen ciphertext attacks under the notion of indistinguishability of ciphertext.

Key Words: Mobile security, Signcryption, Public-key Cryptography.

1. Introduction

Use of mobile personal systems in an open networked environment is very likely to revolutionize the way we use computers. This raises several issues with regard to information security and privacy, system dependability and availability. A networked environment is susceptible to a number of security threats. These include: masquerading, unauthorized use of resources, unauthorized disclosure and flow of information, unauthorized alteration of resources and information, repudiation of actions, unauthorized denial of service. The mobile environment aggravates some of the above security concerns and threats. Because the connection to wireless link may be easy, the security of wireless communication can be compromised much more easily than that of wired communication. The situation gets further complicated if the users are allowed to cress security domains. A mobile system is reachable at any location and at any time. This creates greater concern about privacy issues among the potential users.

Security comes with cost. When communication flows are encrypted or signed digitally, the computational cost will inevitably be added to the total expenses. Therefore, finding efficient security algorithms for secure mobile communications becomes an importance task. In this paper, we propose an offline signcryption scheme where most of computations for signing and encrypting a message are carried out offline and the corresponding online computation is very efficient.

1.1. Previous Work

Digital signatures are used to ensure the authenticity of information and its sender, whereas the information confidentiality is achieved using encryption schemes. Hence to achieve both authenticity and confidentiality both signing and encryption techniques are needed. That is, to secure the message, it is firstly signed and then encrypted. The total computational cost therefore includes the computational costs for performing digital signature and encryption. The notion of signcryption was introduced by Zheng [11], with the goal of achieving greater efficiency than when carrying out the signature and encryption operations separately. Signcryption schemes can achieve both authenticity and confidentiality in public key setting.

The notion of online/offline signature was introduced by Even, Goldreich, and Micali [5]. In this notion, signing phase is broken into two parts. The first part is *offline*, independent of the message to be signed, while the second part is *online* once the message is presented. To ensure that both online signing and verification are efficient, the major computational overhead is shifted to the offline part. Their

* This work is supported by ARC Discovery Grant DP0557493

method uses a one-time signature scheme, i.e., a scheme which can securely sign only a single message. The essence of their method is to apply (online) the ordinary signing algorithm to authenticate a fresh one-time verification key, and then to apply (online) the one-time signing algorithm, which is typically very fast. Since Even, Goldreich, and Micali's online/offline signature has a very inefficient tradeoff between the size of the keys and the complexity of the one-time signing algorithm, it is not practical. Shamir and Tauman [9] proposed an improved version that is more practical.

The notion of online/offline signcryption was introduced by An, Dodis, and Tabin [1]. In their paper, they did not give any concrete method in this work, since they were only interested in general security proofs on signcryption schemes. Like an online/offline signature scheme, an online/offline signcryption should satisfy a basic property, namely efficiency in the online computation. All expensive operations such as exponent computations should be left offline in the first phase of the scheme. It is reasonable to assume that the offline operations are independent of the particular message to be signed, since the message only becomes available at a later stage. The second phase is performed online, once the message is presented. We are interested in online/offline signcryption schemes in which the offline stage is feasible and the online operation, including a symmetric-key encryption and an online signing part, is fast.

1.2. Our Contribution

In this paper, we extend the notion of online/offline signcryption by An, Dodis, and Tabin and provide a concrete scheme. In our scheme, the online part does not require any exponent computations so it is very efficient. We utilize the notion of short signatures [2]; therefore, online signature part of our scheme is very short. We prove that our scheme is secure. It is even more secure then the original signcryption, since to break the scheme, we need to break the combination of the computationally hard problems.

The rest of this paper is organized as follows. In Section II, we define our scheme and give some basic definitions and security requirements. In Section III, we present our scheme and discuss its properties. In Section IV, we provide a concrete proof on the security of our scheme. In Section V, we conclude the paper.

2. Definitions

In this section, we provide definitions of our protocol and its security requirements.

Definition 1 *Our signcryption scheme* SC *is a triple of polynomial-time algorithms* (KeyGen, SigEnc, VerDec), *where*

- $\mathsf{KeyGen}(1^\ell)$ *is a polynomial algorithm that takes as input the security parameter* ℓ *and outputs a pair of keys* $(\mathsf{SDK}, \mathsf{VEK})$. SDK *is the user's sign/decrypt key, which is kept secret, and* VEK *the user's verify/encrypt key, which is made public.*
- SigEnc*, a polynomial algorithm, takes as input the sender* S*'s secret key* SDK *and the receiver* R*'s public key* VEK *and a message* m *from the associated message space* $\mathcal{M}$ *and outputs a signcryption* $s \leftarrow \mathsf{SigEnc}_{\mathsf{SDK},\mathsf{VEK}}(m)$. *This algorithm is split into two parts: online and offline. The offline part does not require the message to be signcrypted and produces an offline signature* S *and a secret key* K *to be used for the offline part. In the online part, The online signature is converted into the fully signature with the given message and the associated secret key related to* K *and the message is encrypted with the key associated with* K.
- VerDec *is a deterministic de-signcryption algorithm that takes as input the signcrypted message* u*, the receiver's secret key* SDK *and the sender* S*'s public key* VEK *and outputs* m *or* $\perp$*, where* $\perp$ *indicates that the message wa not signcrypted properly.*

We require our scheme to be secure against existential forgery under chosen message attacks ($\mathsf{EF\text{-}CPA}$) and adaptive chosen ciphertext attacks under the notion of indistinguishability of ciphertext ($\mathsf{IND\text{-}CCA}$).

The security of the offline signing part of the protocol is based on so-called q-Strong Diffie-Hellman Problem (q-SDHP) introduced in [2], where they used this notion to achieve Strong Existential Unforgeability under chosen message attacks. We will make use this notion in the protection of the sender's authenticity.

Definition 2 *A forger* $\mathcal{A}^{\mathsf{EF\text{-}CPA}}_{q\text{-}\mathsf{SDH}}(t_{q\text{-}\mathsf{SDH}}, q_{q\text{-}\mathsf{SDH}}, \epsilon)$ *breaks an* SC *scheme (offline signing part) if* $\mathcal{A}^{\mathsf{EF\text{-}CPA}}_{q\text{-}\mathsf{SDH}}(t, q_{q\text{-}\mathsf{SDH}}, \epsilon_{q\text{-}\mathsf{SDH}})$ *runs in time at most* $t_{q\text{-}\mathsf{SDH}}$*, makes at most* $q_{q\text{-}\mathsf{SDH}}$ *offline signing queries, and* $\mathsf{Adv\_SC}_{\mathcal{A}^{\mathsf{EF\text{-}CPA}}_{q\text{-}\mathsf{SDH}}}(t_{q\text{-}\mathsf{SDH}}, q_{q\text{-}\mathsf{SDH}}, \epsilon_{q\text{-}\mathsf{SDH}})$ *is at least* $\epsilon_{q\text{-}\mathsf{SDH}}$. *The offline signing is* $(t_{q\text{-}\mathsf{SDH}}, q_{q\text{-}\mathsf{SDH}}, \epsilon_{q\text{-}\mathsf{SDH}})$*-existentially unforgeable under an adaptive chosen message attack against* q-SDH *if no forger* $(t_{q\text{-}\mathsf{SDH}}, q_{q\text{-}\mathsf{SDH}}, \epsilon_{q\text{-}\mathsf{SDH}})$*-breaks it.*

The security of the encryption part in our scheme given in Section 3 refers to the Computational Diffie Hellman (CDH) problem, where the encryption key can be computed if the CDH problem is computable in polynomial time. We required that the encryption part is secure against $\mathsf{IND\text{-}CCA}$ if the CDH problem is hard.

Definition 3 *An adversary* $\mathcal{A}^{\mathsf{IND\text{-}CCA}}_{\mathsf{CDH}}$ $(t_{\mathsf{CDH}}, q_{\mathsf{CDH}}, \epsilon_{\mathsf{CDH}})$*-breaks an* SC *scheme (encryption and online signing) if* $\mathcal{A}^{\mathsf{IND\text{-}CCA}}_{\mathsf{CDH}}$ *runs in time at most* t_{CDH}*, makes at most* q_{CDH} *queries to* CDH *oracle, and* $\mathsf{Adv\_CDH}_{\mathcal{A}^{\mathsf{IND\text{-}CCA}}_{\mathsf{CDH}}}$ *is at least* ϵ_{CDH}*. The encryption and online signing are* $(t_{\mathsf{CDH}}, q_{\mathsf{CDH}}, \epsilon_{\mathsf{CDH}})$*-secure under* IND-CCA *if no adversary* $(t_{\mathsf{CDH}}, q_{\mathsf{CDH}}, \epsilon_{\mathsf{CDH}})$*-breaks it.*

The online signing part in our scheme given in Section 3 is a variant of Schnorr's signature scheme; The security refers to the elliptic curve discrete log (EDL) problem. The security of this kind of problems has been described by Pointcheval and Stern [8] under random oracle assumptions. Their notions are also suitable for elliptic curve settings. We refer this problem to as PC-EDL. The online signing part is secure against EF-CPA if solving PC-EDL in polynomial time is negligible. The security of the online signing part is then defined as follows.

Definition 4 *An adversary* $\mathcal{A}^{\mathsf{EF\text{-}CPA}}_{\mathsf{PC\text{-}EDL}}$ $(t_{\mathsf{PC\text{-}EDL}}, q_{\mathsf{PC\text{-}EDL}}, \epsilon_{\mathsf{PC\text{-}EDL}})$*-breaks a* SC *scheme (online signing) if* $\mathcal{A}^{\mathsf{EF\text{-}CPA}}_{\mathsf{PC\text{-}EDL}}$ *runs in time at most* $t_{\mathsf{PC\text{-}EDL}}$*, makes at most* $q_{\mathsf{PC\text{-}EDL}}$ *queries to* PC-EDL *oracle, and* $\mathsf{Adv\_PC\text{-}EDL}_{\mathcal{A}^{\mathsf{EF\text{-}CPA}}_{\mathsf{PC\text{-}EDL}}}$ *is at least* $\epsilon_{\mathsf{PC\text{-}EDL}}$*. The encryption and online signing are* $(t, q_{\mathsf{PC\text{-}EDL}}, \epsilon_{\mathsf{PC\text{-}EDL}})$*-secure under* EF-CPA *if no adversary* $(t_{\mathsf{PC\text{-}EDL}}, q_{\mathsf{PC\text{-}EDL}}, \epsilon_{\mathsf{PC\text{-}EDL}})$*-breaks it.*

We require our scheme to be secure against EF-CPA on the signing part and IND-CCA on the encryption part. To break the scheme, the adversary has to solve q-SDH, CDH, and PC-EDL problems.

2.1. Bilinear Pairings

In the section, we review some concepts in bilinear pairings provided by Boneh and Franklin [3].

Define two cyclic groups $\mathbb{G}_1, \mathbb{G}_2$. $\mathbb{G}_1$ is an additive group and $\mathbb{G}_2$ is multiplicative group, where both group have a prime order p. Let e be a computable bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$. For $a, b \in \mathbb{Z}_p$ and $P, Q \in \mathbb{G}_1$, we have $e(aP, bQ) = e(P, Q)^{ab}$. We also require non-degeneration $e(P, P) \neq 1$.

Joux and Nguyen [6] showed that an efficiently computable bilinear map e provides an algorithm for solving the Decision Diffie-Hellman problem (DDH). That is given $P, aP, bP, cP \in \mathbb{G}_1$ and $a, b, c \leftarrow \mathbb{Z}_p$, decide whether $c \overset{?}{=} ab \leftarrow \mathbb{Z}_q$. This is because $e(aP, bP) = e(P, cP)$. The computational Diffie-Hellman problem is still hard. Let a, b be chosen from $\mathbb{Z}_p$ at random and P be a generator chosen from $\mathbb{G}_1$ at random. Given (P, aP, bP), it is hard to compute $abP \leftarrow \mathbb{G}_1$.

2.2. The Strong Diffie-Hellman Assumption

In [2], the strong Diffie-Hellman Assumption is referred to as q-SDH, where they utilised a map for two cyclic groups of prime order p, where possibly two cyclic groups are the same. For simplicity, we assume that two cyclic groups both are the same additive group. Therefore, we need to rewrite q-SDH.

Let P be a generator of $\mathbb{G}_1$. The q-SDH problem in $\mathbb{G}_1$ is defined as follows: given a $(q+1)$-tuple $(P, xP, x^2P, \cdots, x^qP)$ as input, output a pair $(c, \frac{1}{x+c}P)$, where $c \leftarrow \mathbb{Z}_p$. An algorithm $\mathcal{A}$ has advantage ϵ in solving q-SDH in $\mathbb{G}_1$ if

$$Pr\left[\mathcal{A}(P, xP, x^2P, \cdots, x^qP = (c, \frac{1}{x+c}P)\right] > \epsilon,$$

where the probability is over the random choice of x in $\mathbb{Z}_p$ and the random bits consumed by $\mathcal{A}$.

Definition 5 *We say that the* (q, t, ϵ)-SDH *assumption holds in* $\mathbb{G}_1$ *if no* t*-time algorithm has advantage at least* ϵ *in solving the* q-SDH *problem in* $\mathbb{G}_1$.

In [2], it is proved that the q-SDH assumption has similar properties to the Strong RSA problem and they therefore view q-SDH as a discrete logarithm analogue of the Strong RSA assumption. A weaker version of the q-SDH assumption was previously used by Mitsunari, Sakai, and Kasahara [7] to construct a traitor tracing system. It was also used in [10] to prove security in their short signature scheme.

3. The Scheme

In this section, we present our online/offline signcryption scheme that satisfies the model introduced in the previous section. Assume that Alice and Bob are the sender and the receiver, respectively. The protocol is described as follows.

- KeyGen. Take ℓ as input and generate Alice's key tuple $(P_{pub1}, P_{pub2}, x, y)$, where $P_{pub1} = xP, P_{pub2} = yP$. $P \in \mathbb{G}_1$ is a public generator and $x, y \leftarrow \mathbb{Z}_q$ are the associated private keys. The same key generator KeyGen generates Bob's key tuple (P_{pubB}, x_b), where $P_{pubB} = x_bP$ and $x_b \in \mathbb{Z}_q$ is the private key of Bob. We have omitted the other key pair, since it is not used by Bob.
- SigEnc. This step is split into two phases, online and offline. The offline phase results in an offline part of the signature and the encryption key for the offline phase. The offline phase produces the offline signature $S = \frac{1}{x+ry}P$ and the keys (y^{-1}, k_1, k_2) for the online phase, where k_1, k_2 are generated from the key generation function $KDF(K)$, where $K = ryP_{pubB}$,

$r \in \mathbb{Z}_p$. In the online phase, the message encryption is done with k_1 and a symmetric-key encryption algorithm such as AES. The resultant ciphertext is $c = E_{k_1}(m)$, $m \leftarrow \mathcal{M} = \mathbb{Z}_p$. The online signature is computed as $\sigma = r - hy^{-1}$, $h = H(m, k_2)$. The signing scheme is actually a variant of the Schnorr's signature scheme. The full signature is thus $s = (c, \sigma, h, S)$.

- VerDec. The verification phase requires the public key of the senders and the public key the receiver. The correct verification requires to verify the equality $e(hP + \sigma P_{pub2} + P_{pub1}, S) = e(P, P)$ and correctly generate $x_b(hP + \sigma P_{pub2}) = K$. Consequently, correctly decrypt the message $D_{k_1}(c) = m$ and verify $h = H(m, k_2)$.

Correctness: The verification of the signcryption protocol is as follows:

$$\begin{aligned} & e(hP + \sigma P_{pub2} + P_{pub1}, S) \\ = \;& e(\frac{1}{x + ry}(hP + (r - hy^{-1})yP + xP), P) \\ = \;& e(P, P). \end{aligned}$$

$$\begin{aligned} & x_b(hP + \sigma P_{pub2}) \\ = \;& x_b hP + x_b(r - hy^{-1})yP \\ = \;& x_b yrP \\ = \;& K. \end{aligned}$$

Third party verification: The signcryption can be verified by a third party, since the verification process does not require the verifier to know the message. By verifying the following equality,

$$e(hP + \sigma P_{pub2} + P_{pub1}, S) \overset{?}{=} e(P, P),$$

the third party is assured of the correctness of the signcryption.

Length of the signing part: The signcryption consists of (c, σ, h, S), where the signing part comprises (σ, h, S). The size of σ, h, and S are $\log_2 p$, 160 bits, and $\log_2 \rho$ respectively; therefore the total length is $\log_2 p + \log_2 \rho + 160$, where ρ is the safe length for $\mathbb{G}_1$. Note that $S = \frac{1}{x+ry}P = \frac{1}{x+y\sigma+h}P$. It is comparable to [2, 4]. We will prove the signing part is secure against EF-CPA.

Performance: In an online/offline scheme, the offline phase must be very efficient and requires minimum computation. Our protocol is designed to meet this requirement. All expensive computations are done in the offline phase. The online phase consists of only simple computations including one hashing, one multiplication, and a symmetric-key encryption.

4. Security

As defined in Section 2, we split the security analysis into three cases in terms of the offline signing part, the computation of K, and the offline signing part. The encryption part of the protocol should be secure against IND-CCA. The security of the encryption part is based on the CDH problem. That is, given P, P_{pub2} and P_{pubB}, compute yx_bP that leads to $K = ryx_bP$, where r is a random number selected from $\mathbb{Z}_p$. The following theorem shows that our scheme is secure against EF-CPA and IND-CCA.

Theorem 1 *Suppose the* CDH-$(t_{\mathsf{CDH}}, q_{\mathsf{CDH}}, \epsilon_{\mathsf{CDH}})$, PC-EDL-$(t_{\mathsf{PC\text{-}EDL}}, q_{\mathsf{PC\text{-}EDL}}, \epsilon_{\mathsf{PC\text{-}EDL}})$*, and* q-SDH-$(t_{q\text{-}\mathsf{SDH}}, q_{q\text{-}\mathsf{SDH}}, \epsilon_{q\text{-}\mathsf{SDH}})$ *assumptions hold. Then our signcryption scheme is* $(t, q_{\mathsf{SC}}, \epsilon)$*-secure against* EF-CPA *and* IND-CCA*, provided that*

$$t \le t_{\mathsf{CDH}} + t_{\mathsf{PC\text{-}EDL}} + t_{q\text{-}\mathsf{SDH}} - O(q_{\mathsf{SC}}),$$

$$q_{\mathsf{SC}} = q_{\mathsf{CDH}} + q_{\mathsf{PC\text{-}EDL}} + q_{q\text{-}\mathsf{SDH}},$$

$$\epsilon = \epsilon_{\mathsf{CDH}} \cdot \epsilon_{\mathsf{PC\text{-}EDL}} \cdot \epsilon_{q\text{-}\mathsf{SDH}}.$$

The proof of Theorem 1 is described in three experiments given in the next three subsections.

Assume $\mathcal{A}$ is a forger that that (t, q_{SC}, ϵ)-breaks the signcryption scheme. For convenience, we will denote by $\mathcal{A}$ all $\mathcal{A}^{\mathsf{IND\text{-}CCA}}_{\mathsf{CDH}}$, $\mathcal{A}^{\mathsf{EF\text{-}CPA}}_{\mathsf{PC\text{-}EDL}}$, and $\mathcal{A}^{\mathsf{EF\text{-}CPA}}_{q\text{-}\mathsf{SDH}}$. $\mathcal{A}$ takes advantage of three separate algorithms $\mathcal{B}_{\mathsf{CDH}}$, $\mathcal{B}_{\mathsf{PC\text{-}EDL}}$, and $\mathcal{B}_{q\text{-}\mathsf{SDH}}$ that, by interacting with $\mathcal{A}$, solve the following problems respectively,

- the CDH problem in time t_{CDH} and probability ϵ_{CDH}.
- the PC-EDL problem in time $t_{\mathsf{PC\text{-}EDL}}$, which is related to the discrete log problem, and probability $\epsilon_{\mathsf{PC\text{-}EDL}}$, and
- the q-SDH in time $t_{q\text{-}\mathsf{SDH}}$ and probability $\epsilon_{q\text{-}\mathsf{SDH}}$.

4.1. Experiment 1: $\mathcal{A}$ interacts with $\mathcal{B}_{\mathsf{CDH}}$

$\mathcal{A}$ interacts with $\mathcal{B}_{\mathsf{CDH}}$, expecting output $Z = yx_bP$. $\mathcal{B}_{\mathsf{CDH}}$ is given (P, P_{pub2}, P_{pubB}).

Query: $\mathcal{A}$ sends q_{CDH} queries to $\mathcal{B}_{\mathsf{CDH}}$. $\mathcal{B}_{\mathsf{CDH}}$ must respond with guesses Z_i, $i = 1, \cdots, q_{\mathsf{CDH}}$.

Response: $\mathcal{B}_{\mathsf{CDH}}$ outputs Z_i, $i = 1, \cdots, q_{\mathsf{CDH}}$ with probability $\Pr(Z = yx_bP | P, P_{pub2}, P_{pubB}) = q_{\mathsf{CDH}}/p$ that it returns the correct value. Here, we assume that $\mathcal{B}_{\mathsf{CDH}}$ does not repeat the values that have been used.

Output: Algorithm $\mathcal{A}$ outputs a forgery Z_* which is randomly selected from q_{CDH} outputs Z_i, $i = 1, \cdots, q_{\mathsf{CDH}}$. $\mathcal{A}$ then picks $r \in \mathbb{Z}_p$ at random and computes $K_* = rZ_*$ as output and split it into k_{1*}, k_{2*} with $KDF(K_*)$.

4.2. Experiment 2: $\mathcal{A}$ interacts with $\mathcal{B}_{\mathsf{PC\text{-}EDL}}$

$\mathcal{A}$ now interacts with $\mathcal{B}_{\mathsf{PC\text{-}EDL}}$, aiming on solving the PC-EDL problem under adaptively chosen-message attack. We utilise Pointcheval and Stern's method based on Forking Lemma [8]. Assume there is a random oracle $\mathcal{O}_H$ in which takes as input k and a message m_i and outputs h_i, where k is a random number. Also assume there is an online signing oracle $\mathcal{O}_\sigma$ that takes as input h_i and outputs an online signing value σ_i.

Query: $\mathcal{A}$ outputs r, k, and a list of distinct q_{SC} messages $m_1, \cdots, m_{q_{\mathsf{PC\text{-}EDL}}} \in \mathcal{M}$, where $q_{\mathsf{PC\text{-}EDL}} < q$.

Response: $\mathcal{B}_{\mathsf{PC\text{-}EDL}}$ must respond with a list of online signing values $\sigma_1, \cdots, \sigma_{q_{\mathsf{PC\text{-}EDL}}}$ on $q_{\mathsf{PC\text{-}EDL}}$ messages from $\mathcal{A}$. $\mathcal{B}_{\mathsf{PC\text{-}EDL}}$ does it with the aid of two oracles:

- Make $q_{\mathsf{PC\text{-}EDL}}$ queries to the random oracle $\mathcal{O}_H$ on input $m_1, \cdots, m_{q_{\mathsf{PC\text{-}EDL}}}$ and k_{2*}. $\mathcal{O}_H$ responds with $h_1, \cdots, h_{q_{\mathsf{PC\text{-}EDL}}}$.
- Then, make $q_{\mathsf{PC\text{-}EDL}}$ queries $m_1, \cdots, m_{q_{\mathsf{PC\text{-}EDL}}}$ to the online signing oracle $\mathcal{O}_\sigma$ that in turn responds with the triple (R_i, h_i, σ_i) for $i = 1, \cdots, q_{\mathsf{PC\text{-}EDL}}$. During the process, the signing oracle also made queries to oracle $\mathcal{O}_H$. R_i is the signing commitment $R_i = r_i P$ for a random r_i.

Output: Given the result and P_{pub2}, $\mathcal{A}$ can verify $\sigma_i P_{pub2} \overset{?}{=} R_i + h_i P$. According to the Forking Lemma [8], two valid signatures can be resulted.

Lemma 1 *Assume that, within time bound $t_{\mathsf{PC\text{-}EDL}}$, $\mathcal{A}$ produces, with probability $\epsilon_{\mathsf{PC\text{-}EDL}} \geq 7q_{\mathsf{PC\text{-}EDL}}/2^{\ell}$, a valid signature (m, R, h, σ). Then there is another machine which has control over $\mathcal{A}$ and produces two valid signatures (m, R, h, σ) and (m, R, h', σ') such that $h \neq h'$, in expected time $t' < 84480 \cdot t_{\mathsf{PC\text{-}EDL}} \cdot q_{\mathsf{PC\text{-}EDL}}/\epsilon_{\mathsf{PC\text{-}EDL}}$.*

Obviously, given $\sigma = r + hy^{-1}$ and $\sigma' = r + h'y^{-1}$, $\mathcal{A}$ can compute $y = \frac{h_i - h_j}{\sigma - \sigma'}$.

Experiment 2 is independent of Experiment 1, since k_{2*} does not influence the result.

4.3. Experiment 3: $\mathcal{A}$ interacts with $\mathcal{B}_{q\text{-}\mathsf{SDH}}$

Assume y is known due to Experiment 2. The online signature can be rewritten as $S = \frac{1}{x+m}P$ for $m = ry$, which is the "weekly secure" scenario of the Boneh and Boye's scheme [2]. Then, we can utilise their notion of the q-SDH problem and the associated lemma:

Lemma 2 *Suppose the $(t', q, \epsilon_{q\text{-}\mathsf{SDH}})$-SDH assumption holds in $\mathbb{G}_1$. Then the online part of the signature in the signcryption scheme is $(t, q_{q\text{-}\mathsf{SDH}}, \epsilon_{q\text{-}\mathsf{SDH}})$-secure against existential forgery under a chosen message attack provided that*

$$t_{q\text{-}\mathsf{SDH}} \leq t' - O(q^2), \quad q_{q\text{-}\mathsf{SDH}} < q.$$

Proof: $\mathcal{A}$, by interacting with $\mathcal{B}_{q\text{-}\mathsf{SDH}}$, solves the q problem in time $t_{q\text{-}\mathsf{SDH}}$ with advantage $\epsilon_{q\text{-}\mathsf{SDH}}$. Algorithm $\mathcal{B}_{q\text{-}\mathsf{SDH}}$ is given a instance $(P, A_1, \cdots, A_q)$ of the q-SDH problem, where $A_i = x^i P \in \mathbb{G}_1$ for $i = 1, \cdots, q$ and some unknown $x \in \mathbb{Z}_p^*$. $\mathcal{B}_{q\text{-}\mathsf{SDH}}$'s goal is to produce a pair $(c, \frac{1}{x+c}P)$ for some $c \in \mathbb{Z}_p^*$. $\mathcal{B}_{q\text{-}\mathsf{SDH}}$ does so by interacting with $\mathcal{A}$ as follows:

Query: $\mathcal{A}$ outputs a list of distinct q_{SC} messages $m_1, \cdots, m_{q_{q\text{-}\mathsf{SDH}}} \in \mathcal{M}$, where $q_{q\text{-}\mathsf{SDH}} < q$. BB sets $q_{q\text{-}\mathsf{SDH}} = q - 1$.

Response: $\mathcal{B}_{q\text{-}\mathsf{SDH}}$ must respond with a public key and the signatures S_i on the $q-1$ messages from $\mathcal{A}$. Let $f(z)$ be the polynomial $f(z) = \prod_{i=1}^{q-1}(z + m_i) = \sum_{i=0}^{q-1} a_i z^i$, where $a_0, \cdots, a_{q-1} \in \mathbb{Z}_p$ are coefficients of the polynomial $f(z)$. Compute:

$$P' = \sum_{i=0}^{q-1} a_i A_i = f(x)P$$

$$P'' = \sum_{i=1}^{q} a_{i-1} A_i = x f(x) P = xP'.$$

The public key given to $\mathcal{A}$ is (P', P''). For each $i = 1, \cdots, q-1$, $\mathcal{B}_{q\text{-}\mathsf{SDH}}$ must generate a signature D_i on m_i. To do so, let $f_i(z)$ be the polynomial $f_i(z) = f(z)/(z+m_i) = \prod_{j=1, j\neq i}^{q-1}(z+m_j)$. We expend f_i and $f_i(z) = \sum_{j=0}^{q-2} b_{ij} z^j$. Compute

$$S_i = \sum_{j=0}^{q-2} b_{ij} A_j = f_i(x)P = \frac{1}{x+m_i}P' \in \mathbb{G}_1.$$

S_i is a valid signature on m_i under the public key (P', P''), since $e(m_i P' + P'', S_i) = e(P', P')$.

Output: $\mathcal{A}$ returns a forgery (m_*, S_*) such that $S_* \in \mathbb{G}_1$ is a valid signature on $m_* \in \mathbb{Z}_q^*$ and $m_* \notin \{m_1, \cdots, m_{q-1}\}$ since there is only one valid signature per message. We have $e(m_* P' + P'', S_*) = e(P', P')$, therefore

$$S_* = \frac{1}{x + m_*}P' = \frac{f(x)}{x + m_*}P.$$

Using long division we write the polynomial f as $f(z) = \gamma(z)(z + m_*) + \gamma_{-1}$ for some polynomial $\gamma(z) = \sum_{i=0}^{q-2} \gamma_i z^i$ and some $\gamma_{-1} \in \mathbb{Z}_p$. Then the rational fraction $f(z)/(y + m_*)$ can be written as

$$f(z)/(y + m_*) = \frac{\gamma_{-1}}{z + m_*} + \sum_{i=0}^{q-2} \gamma_i y^i.$$

$(z+m_*)$ does not divide $f(z)$. Note $\gamma_{-1} \neq 0$, $f(z) = \prod_{i=1}^{q-1}(z+m_i)$ and $m_* \notin \{m_1, \cdots, m_{q-1}\}$. Then, $\mathcal{B}_{q\text{-SDH}}$ computes

$$W = \frac{1}{\gamma - 1}\left(S_* + \sum_{i=0}^{q-1}(-\gamma_i)A_i\right) = \frac{1}{x+m_*}P.$$

and returns (m_*, W) as the solution to the q-SDH instance.

4.4. Security of the Receiver

The security of the receiver is related to the confidentiality of the message. Only the receiver can decrypt the ciphertext c to obtain the message. In other words, only the receiver can compute the decryption key K. The security of the receiver relies on hardness of the CDH problem. To compute $K = x_b(hP + \sigma P_{pub2})$ without knowing the secret key x_b, we need to compute $x_b yP$ from given P, P_{pub2} and P_{pubB}, which has been discussed previously.

5. Conclusion

We have proposed the first online/offline signcryption scheme from bilinear pairings. In our scheme, the computation performed online is very efficient, since it does not require any exponent computations. Our online/offline signcryption can be verified by any third party, because the verification does not take the corresponding message as input. We have also provided a security proof to show that our scheme is secure against IND-CCA and EF-CPA. We showed that the security of our scheme is based on CDH, PC-EDL, and q-SDH. The total time of breaking our scheme is the sum of times required for breaking all these hard problems.

Our scheme is especially suitable for a mobile environment, in particular, for low power mobile devices, because the online security computation part is very efficient.

References

[1] J. H. An, Y. Dodis, and T. Rabin. On the security of joint signature and encryption. In *Advances in Cryptology, Proc. EUROCRYPT 2002,* LNCS 2332, pages 83–107. Springer-Verlag, Berlin, 2002.

[2] D. Boneh and X. Boyen. Short signatures without random oracles. In *Advances in Cryptology, Proc. EUROCRYPT 2004,* LNCS 3027, pages 56–73. Springer Verlag, 2004.

[3] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *Advances in Cryptology, Proc. CRYPTO 2001,* LNCS 2139, pages 213–229. Springer Verlag, 2001.

[4] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *Advances in Cryptology–ASIACRYPT 2001,* LNCS 2248, pages 514–532. Springer Verlag, 2001.

[5] S. Even, O. Goldreich, and S. Micali. On-line/off-line digital signatures. *Journal of Cryptology*, 9:35–67, 1996.

[6] A. Joux and K. Nguyen. Separate decision deffie-hellman from deffie-hellman in cryptographic groups. available from eprint.iacr.org.

[7] S. Mitsunari, R. Sakai, and M. Kasahara. A new traitor tracing. *IEICE Trans. Fundamentals*, E85-A(2):481–484, 2002.

[8] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

[9] A. Shamir and Y. Tauman. Improved online/offine signature schemes. In *Advances in Cryptology, Proc. CRYPTO 2001,* LNCS 2139, pages 355–367. Springer-Verlag, Berlin, 2001.

[10] F. Zhang, R. Safavi-Naini, and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In *Public Key Cryptography (PKC'04), LNCS 2947*, pages 277–290. Springer-Verlag, 2004.

[11] Y. Zheng. Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption). In *Advances in Cryptology – CRYPTO '97 Proceedings*. Springer Verlag, 1997.