Security Arguments for a Class of ID-based Signatures

Jin Zhou [1], YaJuan Zhang[1, 2], YueFei Zhu[1]

[1](Network Engineering Department, Information Engineering University, Zhengzhou 450002, Henan, China.)

[2] (Key Laboratory of Information Engineering, Guangzhou University, Guangzhou 510006, Guangdong, China)

(Jin Zhou, zhoujin820916.jojo@yahoo.com.cn)

Abstract: Provable security based on complexity theory provides an efficient way for providing the convincing evidences of security. In this paper, we present a definition of *generic ID-based signature schemes* (GIBSS) by extending the definition of *generic signature schemes*, and prove the Forking lemma for GIBSS. That is, we provide the Forking lemma for ID-based signature schemes. The theoretical result can be viewed as an extension of the Forking Lemma due to Pointcheval and Stern for ID-based signature schemes, and can help to understand and simplify the security proofs. Then we propose a new and efficient ID-based signature scheme built upon bilinear maps. We prove its security under *k*-CAA computational assumption in the random oracle model.

Key words: ID-based signatures, Forking Lemma, provable security, existential forgery.

1 Introduction

In 1984, Shamir [1] proposed ID-based public key cryptography (ID-PKC) to simplify key management procedures of traditional certificate-based public key infrastructures (PKIs). In ID-PKC, users within a system could use their online identifiers (combined with certain system-wide information) as their public keys; a trusted third party called a private key generator (PKG) generated private keys for users. The direct obtainment of public keys in ID-PKC entirely eliminated the need for certificates and greatly reduced the problems with key management. ID-based public key cryptography has become a good alternative for certificate-based public key setting, especially when efficient key management and moderate security are required.

While ID-based signature schemes rapidly emerged after 1984, it is only in 2001 that bilinear maps over an elliptic curve were used to yield the first entirely practical and secure ID-based encryption scheme (IBE) [2]. Subsequently several ID-based signature schemes based on bilinear maps were proposed, e.g. [3, 4, 5, 6].

A convincing line of research has tried to provide "provable" security for cryptographic schemes. Unfortunately, provable security is at the cost of a considerable loss in terms of efficiency. Another way to achieve some kind of provable security is to identify concrete cryptographic objects such as hash functions with ideal random objects and to use arguments from relativized complexity theory. The model underlying this approach is often called the "random oracle model" that is proposed by Bellare and Rogaway [7]. We use the word "arguments" for security results proved in this model. As usual, these arguments are relative to well-established hard algorithm problems such as factorization or the discrete logarithm. In 2000, Pointcheval and Stern [8] offered some security arguments for standard signature schemes in the random oracle model, and provided the famous Forking lemma for generic signature schemes. The security notion of an ID-based signature scheme is defined to be secure against existential forgery on adaptively chosen message and ID attack (EUF-ACMIA) [3], which is a natural ID-based version of the standard adaptively chosen message attack (EUF-ACM) [9].

Inspired by Pointcheval's results, this paper presents security arguments for *generic ID-based signature schemes* in the random oracle model. The rest of this paper is organized as follows: In Section 2, we recall some preliminary works. In Section 3, we provide the forking lemma for *generic ID-based signature schemes*. In Section 4, we describe a new and efficient ID-based signature scheme using bilinear maps and present the security proof of our scheme. Finally, we end the paper with a brief conclusion.

2 Preliminaries

2.1 Bilinear map groups and related computational problem

Let λ be a security parameter and q be a λ -bit prime number. Let us consider groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ of the same prime order q and let P, Q be generators of respectively $\mathbb{G}_1$ and $\mathbb{G}_2$. We say that $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ are *bilinear map groups* if there exists a bilinear map $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ satisfying the following properties:

1. Bilinearity: $\forall\ (P,Q) \in \mathbb{G}_1 \times \mathbb{G}_2$, $\forall\ \alpha, \beta \in \mathbb{Z}_q$, $\hat{e}(\alpha P, \beta Q) = \hat{e}(P,Q)^{\alpha\beta}$.
2. Non-degeneracy: $\forall\ P \in \mathbb{G}_1$, $\hat{e}(P,Q) = 1$ for all $Q \in \mathbb{G}_2$ iff $P = \mathcal{O}$.
3. Computability: $\forall\ (P,Q) \in \mathbb{G}_1 \times \mathbb{G}_2$, $\hat{e}(P,Q)$ is efficiently computable.
4. There exists an efficient, publicly computable (but not necessarily invertible) isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ such that $\psi(Q) = P$.

The computational assumption for the security of our scheme was previously proposed by S.Mitsunari *et.al* [10]. The problem was called k-CAA (collusion attack algorithm with k traitors) in Mitsunari *et.al*'s traitor tracing scheme.

Definition 1 (*k*-CAA). For an integer k, $x \in_R \mathbb{Z}_q$, and $P \in \mathbb{G}_1$, given

$$\{P, Q = xP, h_1, \ldots, h_k \in \mathbb{Z}_q, \frac{1}{h_1 + x}P, \ldots, \frac{1}{h_k + x}P\}$$

to compute $\frac{1}{h+x}P$ for some $h \notin \{h_1, \ldots, h_k\}$.

3 Forking lemma and generic ID-based signature schemes

In 2000, Pointcheval and Stern presented a notion of generic signature schemes and the famous Forking Lemma. In this paper, we consider a special kind of ID-based signature schemes, which given the input message m, produce a triple (σ_1, h, σ_2), where σ_1 randomly takes its values in a large set, h is the hash value of (m, σ_1) and σ_2 only depends on σ_1 and h for a fixed private key S_{ID}. Each signature is independent of the previous ones. That is, we assume that no σ_1 can appear with probability greater than $2/2^{\lambda}$, where λ is the security parameter. We call this kind of pairing-based schemes as generic ID-based signature schemes (GIBSSs).

Lemma 3.1 [The splitting lemma] [8] let $A \subset X \times Y$ such that $\Pr[(x,y) \in A] \geq \varepsilon$. For any $\alpha < \varepsilon$, define

$$B=\{(x,y)\in X\times Y \,|\, \Pr_{y'\in Y}[(x,y')\in A]\geq \varepsilon-\alpha\} \text{ and } \bar{B}=(X\times Y)\backslash B,$$

Then the following statements hold:

1. $\Pr[B]\geq \alpha$.
2. $\forall (x,y)\in B$, $\Pr_{y'\in Y}[(x,y')\in A]\geq \varepsilon-\alpha$.
3. $\Pr[B|A]\geq \alpha/\varepsilon$.

Lemma 3.2 Let (*Setup*, *Extract*, *Sign*, *Verify*) be a generic ID-based signature scheme with security parameter λ, $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data and which can only ask to the random oracle and private key extraction oracle. We denote by q_H the number of queries that $\mathcal{A}$ can ask to the random oracle, with $q_H>0$. Assume that, within a time bound *T*, $\mathcal{A}$ produces, with probability $\varepsilon\geq 7q_H/2^\lambda$, a valid signature (*m*, *ID*, *r*, *h*, *s*). Then, within time $T'\leq 16q_H T/\varepsilon$, and with probability $\varepsilon'\geq \frac{1}{9}$, a replay of $\mathcal{A}$ outputs two valid signatures (*m*, *ID*, *r*, *h*, *s*) and (m,ID,r,h',s') such that $h\neq h'$.

Proof: We assume that $H(.)$ be the random oracle in the signing phase, and $\mathcal{A}$ asks a polynomial number of questions to the random oracle $H(.)$. These questions are distinct: for instance, $\mathcal{A}$ can store questions and answers in a table. Let $Q_1,\ldots,Q_{q_H}$ be the q_H distinct questions and let $\rho=(\rho_1,\ldots,\rho_{q_H})$ be the list of the q_H answers of $H(.)$, ω which $\mathcal{A}$ has some random information. It is clear that a random choice of $H(.)$ exactly corresponds to a random choice of ρ. Then, for a random choice of (ω,H), with probability ε, $\mathcal{A}$ outputs a valid signature (*m*, *ID*, *r*, *h*, *s*). Since *H* is a random oracle, it is easy to see that the probability for *h* to be equal to *H*(*m*, *r*) is less than $1/2^\lambda$, unless it has been asked during the attack. So, it is likely that the question (*m*, *r*) is actually asked during a successful attack. Accordingly, we define $Ind(\omega,H)$ to be the index of this question: (*m*, *r*)= $Q_{Ind(\omega,H)}$ （we let $Ind(\omega,H)=\infty$ if the question is never asked）. We then define the sets

$$S=\{(\omega,H)\,|\,\mathcal{A}^H(\omega) \text{ succeeds } \& \ Ind(\omega,H)\neq\infty\},$$
$$S_i=\{(\omega,H)\,|\,\mathcal{A}^H(\omega) \text{ succeeds } \& \ Ind(\omega,H)=i\},\ i\in\{1,\ldots,q_H\}.$$

We call *S* the set of the successful pairs (ω,H), and we note that the set $\{S_i\,|\,i\in\{1,\ldots,q_H\}\}$ is a partition of *S*. With those definitions, we find a lower bound for the probability of success, $\nu=\Pr[S]\geq\varepsilon-1/2^\lambda\geq 6\varepsilon/7$. Let *I* be the set consisting of the most likely indices *i*, $I=\{i,\Pr[S_i|S]\geq 1/2q_H\}$. The following lemma claims that, in case of success, the index lies in *I* with probability at least $\frac{1}{2}$.

Lemma 3.3 $\Pr[Ind(\omega,H)\in I|S]\geq 1/2$.

Proof: By definitions of the sets S_i, $\Pr[Ind(\omega,H)\in I|S]=\sum_{i\in I}\Pr[S_i|S]=1-\sum_{i\notin I}\Pr[S_i|S]$.

Since the complement of *I* contains fewer than q_H elements, this probability is at least $1-q_H\times 1/2q_H \geq \frac{1}{2}$. □

We now run the attacker $2/\varepsilon$ times with random ω and random *H*. Since $\nu=\Pr[S]\geq 6\varepsilon/7$, with probability greater than $1-(1-6\varepsilon/7)^{2/\varepsilon}\geq 1-e^{-12/7}$, we get at least one pair (ω,H) in S. It is easily seen that this probability is lower bounded by $1-e^{-12/7}\geq \frac{4}{5}$.

We now apply lemma 3.1 for each integer $i\in I$: we denote by H_i the restriction of *H* to the queries of index strictly less than *i*. Since $\Pr[S_i]\geq \nu/2q_H$, there exists a subset Ω_i of executions such that,

$$\text{for any } (\omega,H)\in\Omega_i,\ \Pr_{H'}[(\omega,H')\in S_i \mid H'_i=H_i]\geq \nu/4q_H,$$

$$\Pr[\Omega_i \mid S_i]\geq 1/2.$$

Since all the subsets S_i are disjoint,

$$\begin{aligned}
&\Pr_{\omega,H}[(\exists i\in I)(\omega,H)\in\Omega_i\cap S_i \mid S]\\
&=\Pr[\bigcup_{i\in I}(\Omega_i\cap S_i)\mid S]\\
&=\sum_{i\in I}\Pr[\Omega_i\cap S_i\mid S]\\
&=\sum_{i\in I}\Pr[\Omega_i\mid S_i]\Pr[S_i\mid S]\\
&\geq(\sum_{i\in I}\Pr[S_i\mid S])/2\\
&\geq \tfrac{1}{4}
\end{aligned}$$

We let β denote the index $Ind(\omega,H)$ corresponding to the successful pair. With probability at least 1/4, $\beta\in I$ and $(\omega,H)\in S_\beta\cap\Omega_\beta$. Consequently, with probability greater than 1/5, the $2/\varepsilon$ attacks have provided a successful pair (ω,H), with $\beta=Ind(\omega,H)\in I$ and $(\omega,H)\in S_\beta$. Furthermore, if we replay the attack, with fixed ω but randomly chosen oracle H' such that $H'_\beta=H_\beta$, we know that $\Pr_{H'}[(\omega,H')\in S_\beta \mid H'_\beta=H_\beta]\geq \nu/4q_H$. Then

$$\begin{aligned}
&\Pr_{H'}[(\omega,H')\in S_\beta \ \text{ and } \rho_\beta\neq\rho'_\beta \mid H'_\beta=H_\beta]\\
&\geq \Pr_{H'}[(\omega,H')\in S_\beta \mid H'_\beta=H_\beta]-\Pr_{H'}[\rho_\beta=\rho'_\beta]\\
&\geq \nu/4q_H-1/2^\lambda\\
&\geq \varepsilon/14q_H
\end{aligned}$$

where $\rho_\beta=H(Q_\beta)$ and $\rho'_\beta=H'(Q_\beta)$. We replay the attack $14q_H/\varepsilon$ times with a new random oracle

H' such that $H'_\beta = H_\beta$. With probability greater than $1-(1-\varepsilon/14q_H)^{14q_H/\varepsilon} > 3/5$, we get another success.

Finally, after less than $2/\varepsilon + 14q_H/\varepsilon$ repetitions of the attack, with probability greater than $\frac{1}{5}\times\frac{3}{5}\geq\frac{1}{9}$, we have obtained two valid signatures (*m*, *ID*, *r*, *h*, *s*) and (m', ID, r', h', s') with $Q_\beta = (m, r) = (m', r')$ and distinct challenges $h = H(Q_\beta) \neq H'(Q_\beta) = h'$.

Theorem 3.4 Let (*Setup*, *Extract*, *Sign*, *Verify*) be a generic ID-based signature scheme with security parameter λ, $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data and which can only ask to the random oracle and private key extraction oracle. We denote by q_H the number of queries that $\mathcal{A}$ can ask to the random oracle, with $q_H > 0$. Assume that, within a time bound *T*, $\mathcal{A}$ produces, with probability $\varepsilon \geq 7q_H/2^\lambda$, a valid signature (*m, ID, r, h, s*). Then there is another machine which has control over $\mathcal{A}$ and produces two valid signatures (*m*, *ID*, *r*, *h*, *s*) and (m, ID, r, h', s') such that $h \neq h'$, in expected time $T' \leq 84480 q_H T/\varepsilon$.

Proof: It is better to see the resulting machine $\mathcal{M}$ as an expected polynomial time Turing machine:

1. $\mathcal{M}$ initializes *j*=0；
2. $\mathcal{M}$ runs $\mathcal{A}$ until it outputs a successful pair $(\omega, H) \in S$ and denotes by N_j the number of calls to $\mathcal{A}$ to obtain this success, and by β the index $Ind(\omega, H)$；
3. $\mathcal{M}$ replays, at most 140 $N_j\alpha^j$ times, $\mathcal{A}$ with fixed ω and random H' such that $H'_\beta = H_\beta$, where $\alpha = \frac{8}{7}$；
4. $\mathcal{M}$ increments *j* and returns to 2, until it gets a successful forking.

For any execution of $\mathcal{M}$, we denote by *J* the last value of *j* and by *N* the total number of calls to $\mathcal{A}$. We want to compute the expectation of *N*. Since $v = \Pr[S]$, and $N_j \geq 1$, then $\Pr[N_j \geq 1/5v] = (1-v)^{1/5v} \geq 3/4$. We define $l = \lceil \log_\alpha q_H \rceil$, so that, 140 $N_j\alpha^j \geq 28q_H/\varepsilon$ for any $j \geq l$, whenever $N_j \geq 1/5v$. Therefore, for any $j \geq l$, when we have a first success in S, with probability greater than 1/4, the index $\beta = Ind(\omega, H)$ is in the set *I* and $(\omega, H) \in S_\beta \cap \Omega_\beta$. Furthermore, with probability greater than 3/4, $N_j \geq 1/5v$. Therefore, with the same conditions as before, that is $\varepsilon \geq 7q_H/2^\lambda$, the probability of getting a successful fork after at most $28q_H/\varepsilon$ iterations at step 3 is greater than $\frac{6}{7}$.

For any $t \geq l$, the probability for *J* to be greater or equal to *t* is less than $(1-\frac{1}{4}\times\frac{3}{4}\times\frac{6}{7})^{t-l}$, which is less than

γ^{t-l}, with $\gamma = \frac{6}{7}$. Furthermore, since $E[N_j] = \sum_{i=1}^{\infty} iv(1-v)^{i-1} = 1/v$,

$$E[N \mid J = t] \le \sum_{j=0}^{j=t} (E[N_j] + 140E[N_j]\alpha^j) \le \frac{141}{v} \times \sum_{j=0}^{j=t} \alpha^j \le \frac{141}{v} \times \frac{\alpha^{t+1}}{\alpha-1}$$

So, the expectation of *N* is

$$\begin{aligned} E[N] &= \sum_t E[N \mid J = t]\Pr[J = t] \\ &\le \frac{141}{v} \sum_t (\frac{\alpha^{t+1}}{\alpha-1}) \Pr[J \ge t] \\ &\le \frac{165}{\varepsilon} [\sum_{t=0}^{t=l-1} (\frac{\alpha^{t+1}}{\alpha-1}) + \sum_{t \ge l} (\frac{\alpha^{t+1}}{\alpha-1}) \gamma^{t-l}] \\ &\le \frac{165\alpha^{l+1}}{\varepsilon(\alpha-1)} [\frac{1}{\alpha-1} + \sum_t (\alpha\gamma)^t] \\ &\le \frac{165\alpha^{l+1}}{\varepsilon(\alpha-1)} (\frac{1}{\alpha-1} + \frac{1}{1-\alpha\gamma}). \end{aligned}$$

Using the definition of l and the values of α and γ, we obtain

$$E[N] \le \frac{165}{\varepsilon} \cdot \frac{64q_H}{7} \cdot (7+49) = \frac{84480q_H}{\varepsilon}.$$

Lemma 3.5 Let (*Setup*, *Extract*, *Sign*, *Verify*) be a generic ID-based signature scheme with security parameter λ, $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data and which can ask to the random oracle , private key extraction oracle and the signing oracle. We denote respectively by q_H and q_S the number of queries that $\mathcal{A}$ can ask to the random oracle and the number of queries that $\mathcal{A}$ can ask to the signer. Assume that, within a time bound *T*, $\mathcal{A}$ produces, with probability $\varepsilon \ge 10(q_S+1)(q_S+q_H)/2^{\lambda}$, a valid signature（*m*, *ID*, *r*, *h*, *s*）. If the triples (*r*, *h*, *s*) can be simulated without knowing the secret key, with an indistinguishable distribution probability, then, a replay of the attacker $\mathcal{A}$, where interactions with the signer are simulated, outputs two valid signatures (*m*, *ID*, *r*, *h*, *s*) and (m, ID, r, h', s') such that $h \ne h'$, within time $T' \le 23q_H T/\varepsilon$ and with probability $\varepsilon' \ge 1/9$.

Proof: As in the previous proof, we let $Q_1, \ldots, Q_{q_H}$ denote the q_H distinct queries to the random oracle, $\rho_1, \ldots, \rho_{q_H}$ the respective answers, and $m_1, \ldots, m_{q_S}$ the q_S queries（possibly all the same）to the signing oracle. Using the simulator, we can simulate the answers of the signer without knowledge of the secret key. For a message m_i, the simulator answers a triple $(r^{(i)}, h^{(i)}, s^{(i)})$. Then, the attacker assumes that $H(m_i, r^{(i)}) = h^{(i)}$ and stores it. The previous proof can be exactly mimicked, expect for the problem added by the simulations: there is some risk of "collisions" of queries, or supposed queries, to the random oracle. Recall that in the definition of generic ID-based signature schemes, we made the assumption that the probability for a "commitment" $r^{(i)}$ to be output by the signing oracle is less than $2/2^{\lambda}$. Then, two kinds of collisions can appear:

1. A pair $(m_i, r^{(i)})$ that the simulator outputs also appears in the list of questions asked to the random oracle by the attacker (some question Q_j). The probability of such an event is less than $q_H q_S 2/2^{\lambda} \le \varepsilon/5$.
2. A pair $(m_i, r^{(i)})$ that the simulator outputs is exactly similar to another pair produced by this simulator (some question $(m_j, r^{(j)})$). The probability of such an event is less than $q_S{}^2/2 \times 2/2^{\lambda} \le \varepsilon/10$.

Altogether, the probability of collisions is less than $3\varepsilon/10$. Therefore,

$$\Pr_{\omega,H}[\mathcal{A} \text{ succeeds and no-collisions}]$$

$$\ge \Pr_{\omega,H}[\mathcal{A} \text{ succeeds}] - \Pr_{\omega,H}[\text{collisions}]$$

$$\ge \varepsilon(1 - \tfrac{3}{10})$$

$$\ge 7\varepsilon/10$$

This is clearly greater than $7q_H/2^{\lambda}$. We can then apply Lemma 3.2. Such a replay succeeds with probability $\varepsilon' \ge \frac{1}{9}$, within time $T' \le 16 q_H T \times 10/7\varepsilon \le 23 q_H T/\varepsilon$.

Theorem 3.6 Let (*Setup, Extract, Sign, Verify*) be a generic ID-based signature scheme with security parameter λ, $\mathcal{A}$ be a probabilistic polynomial time Turing machine whose input only consists of public data and which can ask to the random oracle , private key extraction oracle and the signing oracle. We denote respectively by q_H and q_S the number of queries that $\mathcal{A}$ can ask to the random oracle and the number of queries that $\mathcal{A}$ can ask to the signer. Assume that, within a time bound *T*, $\mathcal{A}$ produces, with probability $\varepsilon \ge 10(q_S + 1)(q_S + q_H)/2^{\lambda}$, a valid signature (*m, ID, r, h, s*). If the triples (*r, h, s*) can be simulated without knowing the secret key, with an indistinguishable distribution probability, then there is another machine which has control over the machine obtained from $\mathcal{A}$ replacing interaction with the signer by simulation and produces two valid signatures (*m, ID, r, h, s*) and (m, ID, r, h', s') such that $h \ne h'$ in expected time $T' \le 120686 q_H T/\varepsilon$.

Proof: The collusion of the attacker $\mathcal{A}$ and the simulator $\mathcal{S}$ defines a Turing machine $\mathcal{B}$ which can only ask to the random oracle and private key extraction. An execution of $\mathcal{B}$ is successful if it outputs a forgery, and if there is no collisions of queries to the random oracle during the process. Then, within a time bound *T*, $\mathcal{B}$ has a probability of success greater than $7\varepsilon/10 \geq 7q_H/2^{\lambda}$. Using Theorem 3.4, within an expected number of steps bounded by 84480 $q_H T/(7\varepsilon/10) \leq 120686 q_H T/\varepsilon$, one can provide two signatures (*m*, *ID*, *r*, *h*, *s*) and (m, ID, r, h', s') such that $h \neq h'$. □

4 A new and efficient ID-based signature scheme

The identity-based signature scheme is specified by four algorithms:

Setup: Given a security parameter $1^{\lambda}(\lambda \in \mathbb{N})$, the parameter generator follows the steps.

- Generate cyclic groups $(\mathbb{G}_1,+)$, $(\mathbb{G}_2,+)$ and $(\mathbb{G}_T,\cdot)$ of prime order $q > 2^{\lambda}$, an isomorphism ψ from $\mathbb{G}_2$ to $\mathbb{G}_1$, and a bilinear pairing $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Pick a random generator $Q \in \mathbb{G}_2$ and set $P = \psi(Q)$.
- Pick a random $s \in \mathbb{Z}_q^*$ and compute $P_{pub} = sP$.
- Pick two cryptographic hash functions $H_1: \{0,1\}^* \to \mathbb{Z}_q^*$ and $H_2: \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{Z}_q$.

The public parameters are $para = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, P, Q, P_{pub}, \hat{e}, \psi, H_1, H_2)$.

Extract: Given an identifier string $ID \in \{0,1\}^*$ of an entity, the algorithm computes $Q_{ID} = (s + H_1(ID))P$ and $S_{ID} = (s + H_1(ID))^{-1}Q$, it returns a private key as S_{ID}.

Sign: In order to sign a message $m \in \{0,1\}^*$, the signer performs as follows.

- Pick a random $x \in \mathbb{Z}_q^*$ and compute $r = xP \in \mathbb{G}_1$.
- Set $h = H_2(m, r) \in \mathbb{Z}_q$.
- Compute $S = (x + h)S_{ID} \in \mathbb{G}_2$.

Verify: The verifier computes $h = H_2(m, r)$, a signature $\sigma = (r, S)$ on a message m of an entity with identity *ID* is valid if and only if $\hat{e}(Q_{ID}, S) = \hat{e}(r, Q)\hat{e}(hP, Q)$.

Obviously, the new scheme is a GIBSS. We now prove that the triples (r, h, S) can be simulated without the knowledge of the signer's secret key.

Lemma 4.1 Given $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, P, Q, P_{pub}, \hat{e}, \psi, H_1, H_2)$ and an identity ID, $Q_{ID} = (s + H_1(ID))P$, $S_{ID} = (s + H_1(ID))^{-1}Q$. The following distributions are the same.

$$\delta = \left\{ (r,h,S) \middle| \begin{array}{c} x \in_R \mathbb{Z}_q^* \\ h \in_R \mathbb{Z}_q \\ r = xP \\ S = (x+h)S_{ID} \end{array} \right\} \text{ and } \delta' = \left\{ (r,h,S) \middle| \begin{array}{c} c \in_R \mathbb{Z}_q^* \\ h \in_R \mathbb{Z}_q \\ S = cQ \\ r = cQ_{ID} - hP \\ r \neq \mathcal{O} \end{array} \right\}$$

Proof: First we choose a triple (α,β,γ) from the set of the signature: let $\alpha\in\mathbb{G}_1^*$, $\beta\in\mathbb{Z}_q$, $\gamma\in\mathbb{G}_2^*$ such that $\hat{e}(Q_{ID},\gamma)=\hat{e}(\alpha,Q)\cdot\hat{e}(\beta P,Q)$. We then compute the probability of appearance of this triple following each distribution of probabilities:

$$\Pr_{\delta}\left[(r,h,S)=(\alpha,\beta,\gamma)\right]=\Pr_{x\neq 0}\begin{bmatrix} xP=\alpha \\ h=\beta \\ (x+h)S_{ID}=\gamma \end{bmatrix}=\frac{1}{q(q-1)}.$$

$$\Pr_{\delta'}\left[(r,h,S)=(\alpha,\beta,\gamma)\right]=\Pr_{r\neq\mathcal{O}}\begin{bmatrix} \alpha=r=cQ_{ID}-hP \\ h=\beta \\ S=cQ=\gamma \end{bmatrix}=\frac{1}{q(q-1)}.$$

That is, we can construct a simulator $\mathcal{M}$, which produces triples (r,h,S) with an identical distribution from those produced by the signer, as follows:

Simulator $\mathcal{M}$**:** For input $(\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,q,P,Q,P_{pub},\hat{e},\psi,H_1,H_2)$ and Q_{ID},

1. $\mathcal{M}$ randomly chooses $c\in\mathbb{Z}_q^*$, $h\in\mathbb{Z}_q$;
2. $\mathcal{M}$ sets $S=cQ$ and $r=cQ_{ID}-hP$;
3. In the (unlikely) situation where $r=\mathcal{O}$, we discard the results and restart the simulation;
4. $\mathcal{M}$ returns the triple (r,h,S).

Theorem 4.2 In the random oracle model, assume that there is an adaptively chosen message and identity attacker F_0 whose input only consists of public data, which has advantage $\varepsilon\geq 10(q_S+1)(q_S+q_{H_2})/q$ against our scheme when running in a time *T* and makes q_i queries to random oracle $H_i(i=1,2)$ and q_S queries to the signing oracle *Sign*(.). Then there exists an algorithm $\mathcal{F}_2$ that is able to solve $q_{H_1}-\mathrm{CAA}$ problem in an expected time $120686q_{H_1}q_{H_2}(T+q_St_1)/\varepsilon$, where t_1 denotes a signing operation.

Proof: From lemma 4.1, we can see that a valid signature of our scheme (r,h,S) can be simulated without knowing the secret key, with an indistinguishable distribution probability. With the Theorem 3.6, using adversary $\mathcal{F}_0$, we can construct another adversary $\mathcal{F}_1$, given public key Q_{ID}, can produces two valid signatures (m,ID,r,h,S) and (m,ID,r,h',S') such that $h\neq h'$ in expected time less than $120686q_{H_2}T/\varepsilon$. Note that, $\mathcal{F}_1$ can not ask to the signing oracle, but can ask to the random oracle and private key extraction oracle.

From the adversary $\mathcal{F}_1$, we can construct a probabilistic algorithm $\mathcal{F}_2$ such that $\mathcal{F}_2$ solves $q_{H_1}-\mathrm{CAA}$ problem. Algorithm $\mathcal{F}_2$ takes as input an instance $\{Q,xQ,h_0,(h_1,(h_1+x)^{-1}Q),\ldots,(h_{q_{H_1}},(h_{q_{H_1}}+x)^{-1}Q)\}$,

$Q \in \mathbb{G}_2, x \in_R \mathbb{Z}_q^*$， $h_i \in_R \mathbb{Z}_q^* \ (i = 0, \ldots, q_{H_1})$ and h_i are both different, it aims at computing $(h_0 + x)^{-1} Q$.

1． $\mathcal{F}_2$ generates $\mathbb{G}_1, \mathbb{G}_T, P, q, \hat{e}, \psi, H_1, H_2$， element P of prime order q， a publicly computable isomorphism ψ from $\mathbb{G}_2$ to $\mathbb{G}_1$， and a bilinear pairing $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ $\mathbb{G}_1$, $P = \psi(Q)$， two hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$ and $H_2 : \{0,1\}^* \times \mathbb{G}_1 \to \mathbb{Z}_q$；

2. $\mathcal{F}_2$ sets $P_{pub} = \psi(xQ) = xP$，

3. $\mathcal{F}_2$ randomly chooses t， $1 \le t \le q_{H_1}$；

4. $\mathcal{F}_2$ runs $\mathcal{F}_1$ with $para = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, P, Q, P_{pub}, \hat{e}, \psi, H_1, H_2)$. During the execution, $\mathcal{F}_2$ emulates $\mathcal{F}_1$'s oracles as following:

- $H_1(.)$： For input ID, $\mathcal{F}_2$ checks if $H_1(ID)$ is defined. If not, he defines

$$H_1(ID) = \begin{cases} h_0 & i = t \\ h_i & i \ne t \end{cases}$$

, and sets $ID_i \leftarrow ID$， $i \leftarrow i + 1$. $\mathcal{F}_2$ returns $H_1(ID)$ to.

- $H_2(.)$： For input (m, r), $\mathcal{F}_2$ checks if $H_2(m, r)$ is defined. If not, $\mathcal{F}_2$ picks randomly $c \in \mathbb{Z}_q$, sets $H_2(m, r) \leftarrow c$. $\mathcal{F}_2$ returns $H_2(m, r)$ to $\mathcal{F}_1$.

- $Extract(.)$： For input ID_i, if $i = t$， $\mathcal{F}_2$ randomly chooses $M \in \mathbb{G}_2$ and enters step 6; if $i \ne t$, $\mathcal{F}_2$ sets $S_i = (x + h_i)^{-1} Q$ to be reply to $\mathcal{F}_1$.

5. If $\mathcal{F}_1$ outputs two valid signatures (m, ID, r, h, S) and (m, ID, r, h', S') such that $h \ne h'$, $\mathcal{F}_2$ can compute as follows: $M = (h - h')^{-1}(S - S')$.

6. $\mathcal{F}_2$ outputs M which is returned as a result of $(h_0 + x)^{-1} Q$ and stops.

From the above construction, when t chosen by $\mathcal{F}_2$ is exactly the index of identity of entity for which $\mathcal{F}_1$ outputs signature, we believe that the result $\mathcal{F}_2$ outputs must be correct, the probability that $\mathcal{F}_2$ outputs correct results is no less than $1/q_{H_1}$ times the probability that $\mathcal{F}_1$ succeeds, $\mathcal{F}_1$'s oracles are both emulated by $\mathcal{F}_2$. So the execution time of $\mathcal{F}_0$ increases from T to $T + q_S t_1$, that is, the execution time of $\mathcal{F}_1$ is no more than

$120686 q_{H_2}(T+q_S t_1)/\varepsilon$. Then we can get the result of $(h_0+x)^{-1}Q$ within the time $120686 q_{H_1} q_{H_2}(T+q_S t_1)/\varepsilon$. □

5 Conclusion

This paper successfully extends the Forking Lemma for ID-based signature schemes. Using the result of this paper, a large class of ID-based signature schemes, which we called *generic ID-based digital signature schemes*, can be proved to be secure easily in the random oracle model. Furthermore, we present a new and efficient ID-based signature scheme and the security proof of our scheme.

References:

[1] Shamir A. Identity-based cryptosystems and signature schemes. In: Advances in Cryptology - CRYPTO'84. Lecture Notes in Computer Science, Vol. 196. Springer-Verlag, Berlin Heidelberg New York, 1984. 47-53.

[2] Boneh D., Franklin M. Identity-based encryption from the Weil pairing. In: Kilian J, eds. Advances in Cryptology- CRYPTO 2001. Lecture Notes in Computer Science, Vol. 2139. Springer-Verlag, Berlin Heidelberg New York, 2001. 213-229.

[3] Cha JC, Cheon JH. An identity-based signature from gap Diffie-Hellman groups. In: Public Key Cryptography - PKC 2003. Lecture Notes in Computer Science, Vol. 2567. Springer-Verlag, Berlin Heidelberg New York, 2003. 18-30.

[4] Hess, F. Efficient identity based signature schemes based on pairings. In: Selected Areas in Cryptography 9th Annual International Workshop, SAC 2002. Lecture Notes in Computer Science, Vol. 2595. Springer-Verlag, Berlin Heidelberg New York, 2003. 310-324.

[5] Yoon HJ, Cheon JH and Kim Y. Batch Verifications with ID-based Signatures. In: Information Security and Cryptology -ICISC 2004, Lecture Notes in Computer Science, Vol. 3506. Springer-Verlag, Berlin Heidelberg New York, 2005. 233-248.

[6] Paterson KG. ID-Based Signatures from Pairings on Elliptic Curves. Electron. Lett., 2002, Vol. 38, No. 18, 1025-1026. http://eprint.iacr/org/2002/004.

[7] Bellare M, Rogaway P. Random Oracles Are Practical: a Paradigm for Designing Efficient Protocols. In Proc. of the 1st CCCS, ACM Press, New York, 1993. 62-73.

[8] Pointcheval D, Stern J. Security arguments for digital signatures and blind signatures. Journal of Cryptology, 2000. 13(3):361-369.

[9] Goldwasser S, Micali S and Rivest R. A digital signature scheme secure against adaptive chosen message attacks. SIAM Journal of Computing, 1988 17(2): 281-308.

[10] Mitsunari S, Sakai R and Kasahara M. A new traitor tracing. IEICE Trans, 2002. Vol. E85-A, No. 481-484.