Yet Another Short Signatures Without Random Oracles from Bilinear Pairings

Fangguo Zhang[1] and Xiaofeng Chen[2]

[1] Department of Electronics and Communication Engineering,
Sun Yat-Sen University, Guangzhou 510275, P.R.China
isdzhfg@zsu.edu.cn

[2] Department of Computer Science,
Sun Yat-Sen University, Guangzhou 510275, P.R.China
isschxf@zsu.edu.cn

Abstract. In recent years, cryptographic protocols based on the bilinear pairings have attracted much attention. One of the most distinguished achievements in this area was the solution to design short signatures. Up to now, there exist two short signature schemes with random oracles and one without random oracles from bilinear pairings. In this paper, we describe another short signature scheme which is existentially unforgeable under a chosen message attack without using random oracles. The security of our scheme depends on a new complexity assumption we call the k+1 square roots assumption. We discuss the relationship between the k+1 square roots assumption and some related problems and give some conjectures. Further more, the k+1 square roots assumption gives even shorter signatures under the random oracles.

Keywords: *Short signature, Bilinear pairings, Standard model, Random oracle*

1 Introduction

It is well known that a signature scheme that produces signatures of length ℓ can have some security level of at most 2^ℓ, which means that given the public key, it is possible to forge a signature on any message in $\mathcal{O}(2^\ell)$. A natural question that arises is how we can concretely construct a signature scheme that can produce shorter signature length whilst maintaining an existential forgery with the same security level.

Short digital signatures are always desirable. They are necessary in situations in which humans are asked to manually key in the signature or when working in low-bandwidth communication environments. They are also useful in general to reduce the communication complexity of any transmission. As noted in [27], when one needs to sign a postcard, it is desirable to minimize the total length of the original message and the appended signature. In the early days, research in this area has been mainly focusing on how to minimize the total length of the message and the appended signature [28, 1] and how to shorten the DSA signature scheme while preserving the same level of security [27]. From Hidden Field Equation (HFE) problem and Syndrome Decoding problem, a number of short signature schemes, such as Quartz [29, 15], McEliece-based signature [16], have been proposed.

Boneh, Lynn and Shacham [10] propsed a totally new approach to design short digital signatures. The resulting signature scheme, referred to as the BLS signature scheme, is based on the Computational Diffie-Hellman (CDH) assumption on elliptic curves with low embedding degree. In BLS signature scheme, with a signature length $\ell = 160$ bits (which is approximately half the size of DSS signatures with the same security level), it provides a security level of approximately $\mathcal{O}(2^{80})$ in the random oracle model. In [36, 5], a more efficient approach to produce a signature of the same length of BLS was proposed. However, its security is based on a stronger assumption.

Provable security is the basic requirement for signature schemes. Currently, most of the practical secure signature schemes were proved in the random oracle model [3]. Security in the random oracle model does not imply security in the real world. The first provably secure signature scheme in the standard model was proposed by Goldwasser *et al.* [22] in 1984. However, the scheme signs a message in a bit-by-bit manner and hence is regarded as not the ideally suitable for applications. Gennaro, Halevi and Rabin [21] , Cramer and Shoup [17] firstly proposed secure signature schemes under the so-called Strong RSA assumption in the standard model and the efficiency of which is suitable for practical use. Later, Camenisch and Lysyanskaya [12] and Fischlin [19] constructed another two provably secure signature schemes under the strong RSA assumption in the standard model. In 2004, Boneh and Boyen [5] proposed a short signature scheme (BB04) from bilinear groups which is existentially unforgeable under a chosen message attack without using random oracles. The security of the scheme depends on a new complexity assumption: *the Strong Diffie-Hellman assumption*. Therefore, it is till a challenge to construct efficient and provably secure signature schemes in the standard model, especially for short signatures.

Contributions. Our main contributions in this paper are:

- We construct a new efficient and provably secure short signature scheme in the standard model from bilinear pairings. The signature size and efficiency of the proposed scheme are same as BB04 scheme. This is the second short signature scheme without random oracles. The security of our scheme depends on a new complexity assumption we call the $k+1$ square roots assumption.
- Under the random oracles, we present the even shorter signatures. It can provide signature whose length is approximately 160 bits. It is comparable to BB04 [5] scheme and ZSS [36] scheme and more efficient than BLS scheme.
- Related to the $k+1$ square roots assumption, we propose and discuss some new mathematical problems and conjectures.

The rest of the paper is organized as follows: The next section contains preliminaries. We briefly review the bilinear pairings and secure signature schemes, and propose the $k+1$ square roots problem and $k+1$ square roots assumption. Section 3 gives the new short signature scheme and its security analysis without random oracles. We also discuss the relationship between our new signature scheme without random oracles and the Chameleon hash signatures. In Section 4 we show that with random oracles the $k+1$ square roots assumption gives even shorter signatures. We give a security proof under the random oracle model. Section 5, we propose and discuss many new mathematical problems and conjectures related to the $k+1$ square roots assumption. Section 6 concludes this paper.

2 Preliminaries

2.1 Bilinear Pairings

In recent years, the bilinear pairings have been found various applications in cryptography and have allowed us to construct some new cryptographic primitives. We briefly review the necessary facts about bilinear pairings using the same notation as [8, 10]:

Let $\mathbb{G}$ be (mutiplicative) cyclic groups of prime order q. Let g be a generator of $\mathbb{G}$.

Definition 1. *A map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ (here $\mathbb{G}_T$ is an additional group such that $|\mathbb{G}| = |\mathbb{G}_T| = q$) is called a bilinear pairing if this map satisfies the following properties:*

1. **Bilinearity:** *For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q$, we have $e(u^a,\ v^b) = e(u,\ v)^{ab}$.*
2. **Non-degeneracy:** $e(g, g) \neq 1$. *In other words, if g be a generator of $\mathbb{G}$, then $e(g,\ g)$ generates $\mathbb{G}_T$.*
3. **Computability:** *There is an efficient algorithm to compute $e(u,\ v)$ for all $u, v \in \mathbb{G}$.*

We say that $\mathbb{G}$ is bilinear group if there exists a group $\mathbb{G}_T$, and a bilinear pairing $e : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_T$ as above. Such groups can be found on supersingular elliptic curves or hyperelliptic curves over finite field, and the bilinear parings can be derived from the Weil or Tate pairing.

2.2 The $k + 1$ Square Roots Assumption

In this subsection, we first introduce a new hard problem which the new signature scheme in this paper is based.

Definition 2 ($k + 1$-SRP). *The $k + 1$* **Square Roots Problem** *in $(\mathbb{G}, \mathbb{G}_T)$ is as follows: For an integer k, and $x \in_R \mathbb{Z}_q$, $g \in \mathbb{G}$, given*

$$\{g, \alpha = g^x, h_1, \ldots, h_k \in \mathbb{Z}_q, g^{(x+h_1)^{\frac{1}{2}}}, \ldots, g^{(x+h_k)^{\frac{1}{2}}}\},$$

to compute $g^{(x+h)^{\frac{1}{2}}}$ for some $h \notin \{h_1, \ldots, h_k\}$.

We say that the $k + 1$-SRP is (t, ϵ)-hard if for any t-time adversary $\mathcal{A}$, we have

$$\Pr\left[\begin{array}{l}\mathcal{A}(g, \alpha = g^x, g^{(x+h_1)^{\frac{1}{2}}}, \ldots, g^{(x+h_k)^{\frac{1}{2}}} | x \in_R \mathbb{Z}_q, g \in \mathbb{G}, h_1, \ldots, h_k \in \mathbb{Z}_q) \\ = g^{(x+h)^{\frac{1}{2}}},\ h \notin \{h_1, \ldots, h_k\}\end{array}\right] < \epsilon.$$

Definition 3 ($k + 1$-SR Assumption). *We say that the $(k + 1, t, \epsilon)$-SR assumption holds in $(\mathbb{G}, \mathbb{G}_T)$ if no t-time algorithm has advantage at least ϵ in solving the k+1-SRP in $(\mathbb{G}, \mathbb{G}_T)$, i.e., $k + 1$-SRP is (t, ϵ)-hard in $(\mathbb{G}, \mathbb{G}_T)$.*

2.3 Secure Signature Schemes

A signature scheme consists of the following four algorithms : a parameter generation algorithm $\mathsf{ParamGen}$, a key generation algorithm KeyGen, a signature generation algorithm Sign and a signature verification algorithm Ver.

There are two specific kinds of attacks against signature schemes: the no-message attack and the known-message attack. In the first scenario the attacker only knows the public key of the signer. In the second one the attacker has access to a list of message-signature pairs. There is a type of the strongest chosen-message attack: the adaptively chosen-message attack, *i.e.*, the attacker has the knowledge of the public key of the signer, and he can ask the signer to sign any message that he wants. He can then adapt his queries according to previous message-signature pairs. The strongest notion of security for signature schemes was defined by Goldwasser, Micali and Rivest [22, 23] as follows:

Definition 4 (Secure signatures [22, 23]). *A signature scheme $\mathcal{S} = <$ ParamGen, KeyGen, Sign, Ver $>$ is existentially unforgeable under an adaptive chosen message attack if it is infeasible for a forger who only knows the public key to produce a valid message-signature pair after obtaining polynomially many signatures on messages of its choice from the signer.*

Formally, for every probabilistic polynomial time forger algorithm $\mathcal{F}$ there does not exist a non-negligible probability ϵ such that

$$\boldsymbol{Adv}(\mathcal{F}) = \Pr\left[\begin{array}{l}\langle pk, sk\rangle \leftarrow \langle \mathsf{ParamGen}, \mathsf{KeyGen}\rangle(1^l);\\ for\ \ i = 1, 2, \ldots, k;\\ m_i \leftarrow \mathcal{F}(pk, m_1, \sigma_1, \ldots, m_{i-1}, \sigma_{i-1}), \sigma_i \leftarrow \mathsf{Sign}(sk, m_i);\\ \langle m, \sigma\rangle \leftarrow \mathcal{F}(pk, m_1, \sigma_1, \ldots, m_k, \sigma_k);\\ m \notin \{m_1, \ldots, m_k\}\ and\ \mathsf{Ver}(pk, m, \sigma) = accept\end{array}\right] \geq \epsilon.$$

Goldwasser *et al.* also realize a signature scheme which satisfies the above security notion. Their scheme has an advantage that it does not use hash functions for message formatting. It is the first secure signature scheme under the standard model.

Here we use the definition of [4] which takes into account the presence of an ideal hash function (the cryptographic hash function is seen as an oracle which produces a random value for each new query), and gives a concrete security analysis of digital signatures.

Definition 5 (Exact security of signatures [4]). *A forger $\mathcal{F}$ is said to (t, q_H, q_S, ϵ)-break the signature scheme $\mathcal{S} = <$ ParamGen, KeyGen, Sign, Ver $>$ via an adaptive chosen message attack if after at most q_H queries to the hash oracle, q_S signatures queries and t processing time, it outputs a valid forgery with probability at least ϵ.*

A signature scheme $\mathcal{S}$ is (t, q_H, q_S, ϵ)-secure if there is no forger who (t, q_H, q_S, ϵ)-breaks the scheme.

3 New Short Signatures Without Random Oracles

3.1 Construction

We describe the new signature scheme as follows:

Let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be the bilinear pairing where $|\mathbb{G}| = |\mathbb{G}_T| = q$ for some prime q. We assume that $|q| \geq 160$. As for the message space, if the signature scheme is intended to be used directly for signing messages, then $|m| = 160$ is good enough, because, given a suitable collision resistant hash function, such as SHA-1, one can first hash a message to 160 bits, and then sign the resulting value. So the messages m to be signed can be regarded as an element in $\mathbb{Z}_q$. In order to give an exact security proof with good bound for the new signature scheme, we limit the message space is $\mathbb{Z}_q[+1] := \{a \in \mathbb{Z}_q |\ a \text{ is a quadratic residue modulo } q\}$. The system parameters are $(\mathbb{G},\ \mathbb{G}_T,\ e,\ q,\ g, \mathbb{Z}_q[+1])$, here $g \in \mathbb{G}$ is a random generator.

Key Generation. Randomly select $x, y \in_R \mathbb{Z}_q^*$, and compute $u = g^x,\ v = g^y$. The public key is $(u,\ v)$. The secret key is (x, y).

Signing: Given a secret key $x, y \in_R \mathbb{Z}_q^*$, and a message $m \in \mathbb{Z}_q[+1]$, pick a random $r \in_R \mathbb{Z}_q^*$, and compute

$$\sigma = g^{(x+my+r)^{\frac{1}{2}}} \in \mathbb{G}.$$

Here $(x+my+r)^{\frac{1}{2}}$ is computed modulo q. When $x+my+r$ is not a quadratic residue modulo q we try again with a different random r. The signature is (σ, r).

Verification: Given a public key $(\mathbb{G},\ \mathbb{G}_T,\ q,\ g,\ u,\ v)$, a message $m \in \mathbb{Z}_q[+1]$, and a signature (σ, r), verify that

$$e(\sigma,\ \sigma) = e(uv^m g^r,\ g).$$

The verification works because of the following equations:

$$\begin{aligned} e(\sigma,\ \sigma) &= e(g^{(x+my+r)^{\frac{1}{2}}},\ g^{(x+my+r)^{\frac{1}{2}}}) \\ &= e(g,\ g)^{(x+my+r)^{\frac{1}{2}} \cdot (x+my+r)^{\frac{1}{2}}} \\ &= e(g,\ g)^{x+my+r} \\ &= e(g^{x+my+r},\ g) \\ &= e(uv^m g^r,\ g) \end{aligned}$$

Notes: From above construction, we can regard the message space as $\mathbb{Z}_q$, and we also can compute the signature as $\sigma = g^{(x+m+yr)^{\frac{1}{2}}} \in \mathbb{G}$. But the security proofs of such schemes are different with the description at section 3.3.

3.2 Efficiency

Using the bilinear groups, there exist three secure signature schemes without random oracles, *i.e.*, BB04 scheme [5], BMS03 scheme [11] and CL04 scheme [13]. BMS03 signature scheme is based on a signature authentication tree with a large branching factor. Compare to BMS03 and CL04 scheme, our scheme has the obvious advantages in all parameters such as the public key and signature lengths and performance.

The new signature scheme requires one computation of square root in $\mathbb{Z}_q^*$ and one exponentiation in $\mathbb{G}$ to sign. For the verification, it needs two pairings and two exponentiations in $\mathbb{G}$. This is same as BB04 scheme.

We note that the computation of the pairing is the most time-consuming in pairing based cryptosystems. Although there have been many papers discussing the complexity of pairings and how to speed up the pairing computation [2, 18, 20], the computation of the pairing still remains time-consuming. Similar to BB04 scheme, some pairings in the proposed signature scheme can be pre-computed and published as part of the signers public key such that there is only one pairing operation in verification. We pre-compute $a = e(u,\ g)$, $b = e(g,\ g)$ and $c = e(v,\ g)$ and published as part of the signers public key. Then, for a message $m \in \mathbb{Z}_q^*$, and a signature (σ, r), the verification can be done as follows:

$$e(\sigma,\ \sigma) \stackrel{?}{=} a \cdot b^m \cdot c^r.$$

So, the verification only needs one pairing and two exponentiations in $\mathbb{G}_T$, and we know that the exponentiations in $\mathbb{G}_T$ tends to be significantly faster than pairing operation.

Signature Length. A signature in the new scheme contains two elements (σ, r), one is in $\mathbb{G}$ and the other is in $\mathbb{Z}_q^*$. When using a supersingular elliptic curve over finite field F_{p^n} with embedding degree $k = 6$ and the modified weil pairing [10], the length of element in $\mathbb{Z}_q^*$ and $\mathbb{G}$ can be approximately $\log_2 q$ bits, therefore the total signature length is approximately $2\log_2 q$ bits. For detail, let $P \in E(F_{p^n})$, $ord(P) = q$, $\mathbb{G} =< P >$. Let ϕ be a distortion map, *i.e.*, an efficiently computable automorphism of $E[q] \cong Z_q \times Z_q$ such that $\phi(P) \notin < P >= \mathbb{G}$. Consider the bilinear pairing

$$\hat{e} : \mathbb{G} \times \mathbb{G} \to \mu_q,$$

defined by

$$\hat{e}(P, Q) = e_w(P, \phi(Q)),$$

here e_w is weil pairing and μ_q is the subgroup of order q in $F_{p^{nk}}^*$.

We can choice the parameter such that the elements in $\mathbb{G}$ are 171-bits strings. So, we obtain a signature whose length is approximately the same as a DSA signature with the same security, but which is provably existentially unforgeable under a chosen message attack without the random oracle model, which is same as BB04. Therefore, this is the second short signature scheme without random oracles.

3.3 Proof of Security

The following theorem shows that the scheme above is existentially unforgeable in the strong sense under chosen message attacks, provided that the $k+1$-SR assumption holds in $(\mathbb{G},\ \mathbb{G}_T)$.

Theorem 1. *Suppose the $(k+1, t', \epsilon')$-SR assumption holds in $(\mathbb{G}, \mathbb{G}_T)$. Then the signature scheme above is (t, q_S, ϵ)-secure against existential forgery under an adaptive chosen message attack provided that*

$$q_S < k+1,\ \epsilon = 2\epsilon' + 4\frac{q_S}{q} \approx 2\epsilon',\ t \le t' - \Theta(q_S T).$$

where T is the maximum time for computing a square root in $\mathbb{Z}_q^*$ and an exponentiation in $\mathbb{G}$.

Proof. To prove the theorem, we will prove: "If there exists a (t, q_S, ϵ)-forger $\mathcal{F}$ using adaptive chosen message attack for the proposed signature scheme, then there exists a (t', ϵ')-algorithm $\mathcal{A}$ solving q_S-SRP (also $k+1$-SRP, if $k+1 > q_S$), where $t' \geq t + \Theta(q_S T)$, $\epsilon' = \frac{\epsilon}{2} - 2\frac{q_S}{q}$."

Assume $\mathcal{F}$ is a forger that (t, q_S, ϵ)-breaks the signature scheme. We construct an algorithm $\mathcal{A}$ that, by interacting with $\mathcal{F}$, solves the q_S-SRP in time t' with advantage ϵ'.

Suppose $\mathcal{A}$ is given a challenge – a random instance of q_S-SRP:

" For an integer q_S, and $x \in_R \mathbb{Z}_q$, $g \in \mathbb{G}$, given

$$\{g,\ \alpha = g^x,\ h_1, \ldots, h_{q_S} \in \mathbb{Z}_q,\ g^{(x+h_1)^{\frac{1}{2}}}, \ldots, g^{(x+h_{q_S})^{\frac{1}{2}}}\},$$

to compute $g^{(x+h)^{\frac{1}{2}}}$ for some $h \notin \{h_1, \ldots, h_{q_S}\}$."

Next, we describe how the algorithm $\mathcal{A}$ to solve the q_S-SRP by interacting with $\mathcal{F}$. The approach is similar to BB04 [5] and [35]. We distinguish between two types of forgers that $\mathcal{F}$ can emulate. Let $(\mathbb{G},\ \mathbb{G}_T,\ q,\ g, u,\ v)$ be the public key given to forger $\mathcal{F}$ where $u = g^x$ and $v = g^y$. Suppose $\mathcal{F}$ asks for signatures on messages $m_1, m_2, \cdots, m_{q_S} \in \mathbb{Z}_q^*$ and is given signatures $(r_i,\ \sigma_i)$ on these messages for $i = 1, \cdots, q_S$. Let $h_i = m_i y + r_i$ and let (m, r, σ) be the forgery produced by $\mathcal{F}$. Denote two types of forger $\mathcal{F}$ as:

Type-1 Forger which either makes query for $m_i = -x$, or outputs a forgery where $my + r \notin \{h_1, h_2, \cdots, h_{q_S}\}$.

Type-2 Forger which both never makes query for message $m = -x$, and outputs a forgery where $my + r \in \{h_1, h_2, \cdots, h_{q_S}\}$.

$\mathcal{A}$ plays the role of the signer, it produces a forgery for the signature scheme as follows:

Setup: $\mathcal{A}$ is given $g,\ \alpha = g^x$, with q_S known solutions $(h_i \in \mathbb{Z}_q,\ s_i = g^{(x+h_i)^{\frac{1}{2}}} \in \mathbb{G})$ for random h_i $(i = 1, \cdots, q_S)$. $\mathcal{A}$ picks random $y \in \mathbb{Z}_q$ and a bit $b_{mode} \in \{1, 2\}$ randomly. If $b_{mode} = 1$, $\mathcal{A}$ publishes the public key $PK_1 = (\mathbb{G},\ \mathbb{G}_T,\ q,\ g, u,\ v)$, here $u = \alpha,\ v = g^y$. If $b_{mode} = 2$, $\mathcal{A}$ publishes the public key $PK_2 = (\mathbb{G},\ \mathbb{G}_T,\ q,\ g, u,\ v)$, here $u = g^y,\ v = \alpha$. In $\mathcal{F}$'s view, both PK_1 and PK_2 are valid public keys for the signature scheme.

Simulation: The forger $\mathcal{F}$ can issue up to q_S signature queries in an adaptive fashion. To respond these signature queries, $\mathcal{A}$ maintains a list H-list of tuples (m_i, r_i, h_i) and a query counter l which is initially set to 0.

Upon receiving a signature query for m_i, $\mathcal{A}$ increments l by one, and checks if $l > q_S$. If $l > q_S$, it neglects further queries by $\mathcal{F}$ and terminate $\mathcal{F}$. Otherwise, checks if $g^{-m_i} = u$. If so, then $\mathcal{A}$ just obtained the private key for the public key $PK = (\mathbb{G},\ \mathbb{G}_T,\ q,\ g, u,\ v)$ it was given, which allows it to forge the signature on any message of its choice. At this point $\mathcal{A}$ successfully terminates the simulation.

Otherwise, if $b_{mode} = 1$, set $r_i = h_i - m_i y \in \mathbb{Z}_q$. In the very unlikely event that $r_i = 0$, $\mathcal{A}$ reports failure and aborts. Otherwise, $\mathcal{A}$ gives $\mathcal{F}$ the signature $(r_i,\ \sigma_i = s_i)$ This is a valid signature on m_i under the public key $PK_1 = (\mathbb{G},\ \mathbb{G}_T,\ q,\ g, u,\ v)$ since r_i is uniform in $\mathbb{Z}_q$

and

$$e(\sigma_i,\ \sigma_i) = e(g^{(x+h_i)^{\frac{1}{2}}},\ g^{(x+h_i)^{\frac{1}{2}}}) = e(ug^{h_i},\ g) = e(ug^{r_i+m_i y},\ g) = e(uv^{m_i}g^{r_i},\ g).$$

If $b_{mode} = 2$, , set $r_i = m_i h_i - y \in \mathbb{Z}_q$. If $r_i = 0$, $\mathcal{A}$ reports failure and aborts. Otherwise, $\mathcal{A}$ returns $(r_i,\ \sigma_i = s_i^{\sqrt{m_i}})$ as answer (**This is why we limit the message space is** $\mathbb{Z}_q[+1]$). This is a valid signature on m_i for PK_2 because r_i is uniform in $\mathbb{Z}_q$ and

$$\begin{aligned} e(\sigma,\ \sigma) &= e(g^{(x+h_i)^{\frac{1}{2}}\sqrt{m_i}},\ g^{(x+h_i)^{\frac{1}{2}}\sqrt{m_i}}) \\ &= e(g^{m_i h_i}v^{m_i},\ g) \\ &= e(g^{y+r_i}v^{m_i},\ g) \\ &= e(uv^{m_i}g^{r_i},\ g) \end{aligned}$$

$\mathcal{A}$ adds the tuple $(m_i,\ r_i,\ v^{m_i}g^{r_i})$ to H-list.

Reduction: Eventually, the forger $\mathcal{F}$ returns a forgery (m, r, σ), where (r, σ) is a valid forgery distinct from any previously given signature on message m. Note that by adding dummy queries as necessary, we may assume that $\mathcal{F}$ made exactly q_S signature queries. Let $W \leftarrow v^m g^r$. Algorithm $\mathcal{A}$ searches the H-list for a tuple whose rightmost component is equal to W. Then according to two types of forger $\mathcal{F}$, we denote the following events as:

F1: **(Type-1 forgery:)** No tuple of the form $(\cdot,\ \cdot,\ W)$ appears on the H-list.
F2: **(Type-2 forgery:)** The H-list contains at least one tuple (m_j, r_j, W_j) such that $W_j = W$.

Denote E1 to be the event $b_{mode} = 1$ (i.e., $\mathcal{F}$ produced a type-1 forgery, or $\mathcal{F}$ made a signature query for a message m_i such that $g^{-m_i} = u$.) and denote E2 to be the event $b_{mode} = 2$. We claim that $\mathcal{A}$ can succeed in breaking the signature scheme if $(E1 \wedge F1) \vee (E2 \wedge F2)$ happens.

Case 1. If $u = g^{-m_i}$, then $\mathcal{A}$ has already recovered the secret key of its challenger, $\mathcal{A}$ can forge signature on any message of his choice. We assume that $\mathcal{F}$ produced a type-1 forgery (m, r, σ). Since the forgery is valid, we have

$$e(\sigma,\ \sigma) = e(uv^m g^r,\ g) = e(ug^{my+r},\ g).$$

Let $h = my + r$. So, the forgery (m, r, σ) provides a new $q_S - SRP$ solution (h, σ).

Case 2. Since $v = \alpha = g^x$, then we know that there exists a pair $v^{m_j}g^{r_j} = v^m g^r$. Since $(m, r) \neq (m_j, r_j)$, otherwise it is not regarded as a forgery, so, $m \neq m_j,\ r \neq r_j$. Therefor, $\mathcal{A}$ can compute $x = \frac{r_j - r}{m - m_j}$ which also enables $\mathcal{A}$ to recover the secret key of its challenger. He can now forge a signature on any message of its choice.

Any valid forgery (m, r, σ) will give a new $q_S - SRP$ solution under at least one of the 2 above reductions.

This completes the description of Algorithm $\mathcal{A}$. A standard argument shows that if $\mathcal{A}$ does not abort, then, from the viewpoint of $\mathcal{F}$, the simulation provided by $\mathcal{A}$ is indistinguishable from a real attack scenario. Since the simulations are perfect, $\mathcal{F}$ cannot guess which reduction the simulator is using. Therefore, $\mathcal{F}$ produces a valid forgery in time t with probability at least ϵ.

Since E1 and F1 are independent with uniform distribution, $Pr[E1 \vee E2] = 1$ and $Pr[F1 \vee F2] = 1$, the probability that $\mathcal{A}$ succeeds is $Pr[(E1 \wedge F1) \vee (E2 \wedge F2)] = \frac{1}{2}$.

Next we bound the probability that $\mathcal{A}$ dos not abort. From above description of $\mathcal{A}$ we know that $\mathcal{A}$ aborts if

- At $E1 \wedge F1$, only if $r_i = 0$, *i.e.*, $m_i y = h_i$. For given y, this happens with probability at most $\frac{q_S}{q}$.
- or at $E2 \wedge F2$, only if $r_i = 0$, *i.e.*, $m_i h_i = y$. For given y, this happens with probability at most $\frac{q_S}{q}$.

So, $\mathcal{A}$ succeeds with probability at least $\frac{\epsilon}{2} - 2\frac{q_S}{q}$.

Let T be the maximum time for a computing square root in $\mathbb{Z}_q^*$ and an exponentiation in $\mathbb{G}$. The running time of $\mathcal{A}$ is $t' \geq t + \Theta(q_S T)$. This complete the proof. $\square$

3.4 Relation to Chameleon Hash Signatures and On-line/Off-line Signatures

Chameleon signatures, introduced by Krawczyk and Rabin [25], are based on well established hash-and-sign paradigm, where a *chameleon hash function* is used to compute the cryptographic message digest. A chameleon hash function is a trapdoor one-way hash function, which prevents everyone except the holder of the trapdoor information from computing the collisions for a randomly given input. Chameleon signatures simultaneously provide the properties of non-repudiation and non-transferability for the signed message, *i.e.*, the designated recipient is capable of verifying the validity of the signature, but cannot disclose the contents of the signed information to convince any third party without the signer's consent.

Similar to the discussion in BB04[5], the $my + r$ component in our signature scheme provides us with the functionality of a chameleon hash too: given m, we can choose r so that $my + r$ maps to some predefined value of our choice. This makes it possible to handle the chosen message attack. Embedding the hash $my + r$ directly in the signature scheme results in a much more efficient construction than using an explicit chameleon hash (which requires additional exponentiations). Therefore, our new signature scheme is a chameleon signature scheme.

Shamir and Tauman [33] showed that chameleon hash function can be used to develop a new paradigm called hash-sign-switch, which can convert any signature scheme into a highly efficient on-line/off-line signature scheme. It is easy to convert our new signature scheme into a highly efficient on-line/off-line signature scheme:

- **Key Generation.** This is the same as the scheme given in Section 3.1.
- **Signing:** This step is split into two phases, online and offline.

  Offline phase.
  The signer selects $r \in Z_p$ and computes:

  $$\sigma = g^{(x+r)^{\frac{1}{2}}},$$

  Online phase.

For any message m, the signer computes:

$$r' = r - my,$$

Publish (r', σ) as the signature on m.

– **Verification:** Given a public key $(\mathbb{G},\ \mathbb{G}_T,\ q,\ g,\ u,\ v)$, a message m, and a signature (σ, r'), verify that

$$e(\sigma,\ \sigma) = e(uv^m g^{r'},\ g).$$

4 Shorter Signatures with Random Oracles

We present a more efficient short signature scheme based on $q_S - SRP$ in the random oracle model. The proposed new short signature scheme with random oracle is described as follows:

The system parameters are $(\mathbb{G},\ \mathbb{G}_T,\ e,\ q,\ g, I)$, here $g \in \mathbb{G}$ is a random generator and I is the upper bound of i used in the signing and verification phase.

Key Generation. Randomly select $x \in_R \mathbb{Z}_q^*$, and compute $u = g^x$. The public key is u. The secret key is x.

Signing: Given a secret key x, and a message m, computes $\sigma = g^{(H(m||i)+x)^{\frac{1}{2}}}$. The signature σ is computed for i starting at 0 and increasing by 1 at each try, until $H(m||i) + x$ is a quadratic residue modulo q.

Verification: Given a public key $(\mathbb{G},\ \mathbb{G}_T,\ e,\ q,\ g,\ u, I)$, a message $m \in \mathbb{Z}_q^*$, and a signature σ, verify that

$$e(\sigma,\ \sigma) = e(g^{H(m||i)}u,\ g).$$

Here i starting at 0 and increasing by 1 at each try, until $H(m||i) + x$ is a quadratic residue modulo q.

The verification works because of the following equations:

$$\begin{aligned}
e(\sigma,\ \sigma) &= e(g^{(x+H(m||i))^{\frac{1}{2}}},\ g^{(x+H(m||i))^{\frac{1}{2}}})\\
&= e(g,\ g)^{(x+H(m||i))^{\frac{1}{2}}\cdot(x+H(m||i))^{\frac{1}{2}}}\\
&= e(g,\ g)^{x+H(m||i)}\\
&= e(g^{x+H(m||i)},\ g)\\
&= e(ug^{H(m||i)},\ g)
\end{aligned}$$

The failure probability can be made arbitrarily small by picking an appropriately large I. For each i, the probability that $H(m||i) + x$ leads to a quadratic residue modulo q is approximately $1/2$. Hence, the probability that a given message m will be failure is $\frac{1}{2^I}$.

We pre-compute $a = e(u,\ g)$ and $b = e(g,\ g)$ and publish them as part of the signers public key. Then, for a message $m \in \mathbb{Z}_q^*$, and a signature σ, the verification can be done as follows:

$$e(\sigma,\ \sigma)/b \stackrel{?}{=} a^{H(m||i)}.$$

This signature scheme can provide the same signature length as BLS scheme. We compare this signature scheme with the BLS scheme from the view point of computation overhead. The key and signature generation times are comparable to BLS signatures. The verification time is faster since verification requires only one pairing and one exponentiation if the signature is (σ, i). If the signature is only σ, then this scheme will need one pairing and many exponentiations in $\mathbb{G}_T$ due to the pre-computation of $a = e(u,\ g)$ and $b = e(g,\ g)$, but BLS scheme will need more pairings.

About the security of proposed signature scheme against an adaptive chosen message attack, we have the following theorem:

Theorem 2. *If there exists a (t, q_H, q_S, ϵ)-forger $\mathcal{F}$ using adaptive chosen message attack for the proposed signature scheme, then there exists a (t', ϵ')-algorithm $\mathcal{A}$ solving $q_H - k$-SRP (for a constant $k \in \mathbb{Z}^+$), where*

$$t = t',\ \epsilon' \geq \prod_{j=0}^{q_S-1} \frac{q_H - k - j}{q_H - j} \cdot \frac{k}{q_H} \cdot \epsilon.$$

Especially, there exists a $(t' = t,\ \epsilon' \geq \frac{q_S}{q_H^2} \cdot \epsilon)$-algorithm $\mathcal{A}$ solving $q_H - 1$-SRP.

Proof. In the proposed signature scheme, before signing a message m, we need to make a query $H(m||i)$. We ignore the case that $H(m||i)$ is not a quadratic residue modulo q. In other words, we assume that for any hash query, the hash oracle will give a right response. Our proof is in random oracle model (the hash function is seen as a random oracle, *i.e.*, the output of the hash function is uniformly distributed).

Suppose that a forger $\mathcal{F}$ (t, q_H, q_S, ϵ)-break the signature scheme using an adaptive chosen message attack. We will use $\mathcal{F}$ to construct an algorithm $\mathcal{A}$ to solve $q_H - 1$-SRP.

Suppose $\mathcal{A}$ is given a challenge:

“ For integer q_H and k, and $x \in_R \mathbb{Z}_q$, $g \in \mathbb{G}$, given

$$\{g,\ \alpha = g^x,\ h_1, \ldots, h_{q_H-k} \in \mathbb{Z}_q,\ g^{(x+h_1)^{\frac{1}{2}}}, \ldots, g^{(x+h_{q_H-k})^{\frac{1}{2}}}\},$$

to compute $g^{(x+h)^{\frac{1}{2}}}$ for some $h \notin \{h_1, \ldots, h_{q_H-k}\}$.”

Now $\mathcal{A}$ plays the role of the signer and sets the public key be $u = \alpha$. $\mathcal{A}$ will answer hash oracle queries and signing queries itself. We assume that $\mathcal{F}$ never repeats a hash query or a signature query.

- S1 $\mathcal{A}$ prepares q_H responses $\{w_1, w_2, \ldots, w_{q_H}\}$ of the hash oracle queries, $h_1, \ldots, h_{q_H-k}$ are distributed randomly in this response set.
- S2 $\mathcal{F}$ makes a hash oracle query on m_j for $1 \leq j \leq q_H$. $\mathcal{A}$ sends w_j to $\mathcal{F}$ as the response of the hash oracle query on m_j.
- S3 $\mathcal{F}$ makes a signature oracle query for w_j. If $w_i = h_j$, $\mathcal{A}$ returns $g^{(x+h_j)^{\frac{1}{2}}}$ to $\mathcal{F}$ as the response. Otherwise, $\mathcal{A}$ reports failure and aborts.
- S4 Eventually, $\mathcal{F}$ halts and outputs a message-signature pair (m, σ). Here the hash value of m is some w_l and $w_l \notin \{h_1, \ldots, h_{q_H-k}\}$. Since (m, σ) is a valid forgery and $H(m||i) = w_l$, it satisfies:

$$e(\sigma,\ \sigma) = e(g^{H(m||i)}u,\ g).$$

So, $\sigma = g^{(x+w_l)^{\frac{1}{2}}}$. $\mathcal{A}$ outputs (w_l, σ) as a solution to $\mathcal{A}$'s challenge.

Algorithm $\mathcal{A}$ simulates the random oracles and signature oracle perfectly for $\mathcal{F}$. $\mathcal{F}$ cannot distinguish between $\mathcal{A}$'s simulation and real life because the hash function behaves as a random oracle. Therefore $\mathcal{F}$ produces a valid forgery for the signature scheme with probability at least ϵ.

Now, we bound the probability $\mathcal{A}$ dos not abort. In step S3, the success probability of $\mathcal{A}$ is $\frac{q_H-k}{q_H}$, so, for all signature oracle queries, $\mathcal{A}$ will not fail with probability

$$\rho \geq \prod_{j=0}^{q_S-1} \frac{q_H-k-j}{q_H-j}$$

(if $\mathcal{F}$ only makes $s(\leq q_S)$ signature oracle queries, the success probability of $\mathcal{A}$ is $\prod_{j=0}^{s-1} \frac{q_H-k-j}{q_H-j}$). Hence, after the algorithm $\mathcal{A}$ finished the step S4, the success probability of $\mathcal{A}$ is:

$$\epsilon' \geq \prod_{j=0}^{q_S-1} \frac{q_H-k-j}{q_H-j} \cdot \frac{k}{q_H} \cdot \epsilon.$$

Especially, if let $k=1$, then the success probability of $\mathcal{A}$ is:

$$\epsilon' \geq \frac{q_S}{q_H^2} \cdot \epsilon.$$

The running time of $\mathcal{A}$ is equal to the running time of $\mathcal{F}$ $t' = t$. $\square$

Another most impressive application of pairings to cryptography is the identity-based encryption scheme [8]. The concept of ID-based cryptosystem was first introduced by Shamir [32]. The basic idea of ID-based cryptosystem is to use the identity information of a user as his public key. As noted in [8], there is a relationship between the short signature schemes and the ID-based public key setting from bilinear pairing, that is the signing process in the short signature scheme can be regarded as the private key extract process in the ID-based public key setting. Therefore, how to construct ID-based cryptosystem using the new short signature, such as ID-based encryption schemes [8, 6], ID-based signature schemes[14, 24, 30], *etc.*, is an interesting topic.

5 Some New Mathematical Problems

Before describe some mathematical problems, we need the following notions from complexity theory.

- We say problem **A** is polynomial time reducible to problem **B**, denoted by $\mathbf{B} \Longrightarrow \mathbf{A}$, if there exists a polynomial time algorithm $\mathcal{R}$ for solving problem **A** that makes calls to a subroutine for problem **B**. In this case, we also say the problem **B** is *harder* than the problem **A**.
- We say that **A** and **B** are polynomial time equivalent if **A** is polynomial time reducible to **B** and **B** is polynomial time reducible to **A**.

Now we describe two well studied problems in the group $(\mathbb{G}, \cdot)$.

– **Discrete Logarithm Problem (DLP):** Given two group elements g and h, find an integer $n \in \mathbb{Z}_q^*$, such that $h = g^n$ whenever such an integer exists.
– **Computational Diffie-Hellman Problem (CDHP):** For $a, b \in \mathbb{Z}_q^*$, given g, g^a, g^b, compute g^{ab}.

There are two variations of CDHP:

– **Inverse Computational Diffie-Hellman Problem (Inv-CDHP):** For $a \in \mathbb{Z}_q^*$, given g, g^a, to compute $g^{a^{-1}}$.
– **Square Computational Diffie-Hellman Problem (Squ-CDHP):** For $a \in \mathbb{Z}_q^*$, given g, g^a, to compute g^{a^2}.

Due to the results of [26, 31], we have the following theorem:

Theorem 3. *CDHP, Inv-CDHP and Squ-CDHP are polynomial time equivalent.*

In the following, we definite a new problem in $\mathbb{G}$ that we call **Reverse Square Computational Diffie-Hellman Problem (RSCDHP)**, which is closely related to the proposed signature scheme with hash function.

Definition 6 (RSCDHP). *For $y \in \mathbb{Z}_q^*$, given g, g^{y^2}, to compute g^y.*

Theorem 4. *The new signature scheme with hash function is secure under no-message attack if RSCDHP is hard, i.e., if there exists a (t, q_H, ϵ)-forger $\mathcal{F}$ against no-message attack for new scheme, then there exists an (t', ϵ')-algorithm $\mathcal{A}$ solving RSCDHP, where $t' = t, \epsilon' = \frac{1}{q_H}\epsilon$.*

Proof. Suppose that a forger $\mathcal{F}$ via no-message attack (t, q_H, ϵ)-breaks the proposed scheme. We will use $\mathcal{F}$ to construct an attack algorithm $\mathcal{A}$ to solve RSCDHP. Suppose that $\mathcal{A}$ is given a challenge:
" For $y \in \mathbb{Z}_q^*$, given g, g^{y^2}, to compute g^y."

$\mathcal{A}$ chooses $t \in Z_q^*$ at random, then $\mathcal{A}$ runs $\mathcal{F}$ with the system parameter $(\mathbb{G}, \mathbb{G}_T, e, q, g, I)$, the public key is $u = g^{y^2}/g^t$. $\mathcal{F}$ makes hash oracle queries during its execution. $\mathcal{A}$ picks an integer i_0 from $\{1, \cdots, q_H\}$ at random.

Now, suppose $\mathcal{F}$ makes a hash oracle query on m_i for $1 \leq i \leq q_H$. If $i = i_0$, then $\mathcal{A}$ returns t as a hash value of m_{i_0}. Otherwise, $\mathcal{A}$ chooses $h_i \in Z_q^*$ and returns it as the hash value of m_i. Eventually $\mathcal{F}$ halts and outputs a message-signature pair (m, σ). Without loss of generality we may assume that $\mathcal{F}$ has requested the hash query m before. Suppose $m = m_i$ for some i. If $i \neq i_0$, then $\mathcal{A}$ outputs "failure" and halts. Otherwise, $\mathcal{A}$ outputs σ as a solution of RSCDHP given by g and g^{y^2}. Since (m, σ) is a valid forgery and $\mathcal{H}(m) = t$, it satisfies:

$$e(\sigma,\ \sigma) = e(ug^{\mathcal{H}(m)},\ g) = e(g^{y^2}/g^t \cdot g^t,\ g) = e(g^{y^2},\ g).$$

The running time of $\mathcal{A}$ is equal to the running time of $t' = t$. The the success probability of $\mathcal{A}$ is: $\epsilon' = \frac{1}{q_H}\epsilon$. $\square$

It is not hard to prove that

Theorem 5. *RSCDHP* $\Longrightarrow$ *1-RSP* $\Longrightarrow$ *2-RSP* $\Longrightarrow \cdots \Longrightarrow$ *k-RSP* $\Longrightarrow$ *k+1-RSP.*

Similar to the Square Computational Diffie-Hellman Problem and Reverse Square Computational Diffie-Hellman Problem, we have

Definition 7 (k+1 Exponent Problem [36]). *Given* $k+1$ *values* $< g, g^y, g^{y^2}, \ldots, g^{y^k} >$, *compute* $g^{y^{k+1}}$.

Definition 8 (k-RSCDH problem). *For* $y \in \mathbb{Z}_q^*$, *given* $g, g^{y^2}, g^{y^3}, \ldots, g^{y^k}, g^{y^{k+1}}$ *to compute* g^y.

We present some open problems and conjectures below for the first time:

Conjecture 1 *k-RSP and k-RSCDHP are polynomial time equivalent.*

Motivated by the signature scheme we also formulate a strong form of the conjecture.

Conjecture 2 *RSCDHP is harder than SCDHP. Especially, if the order* q *of* $\mathbb{G}$ *is prime, RSCDHP and SCDHP are polynomial time equivalent.*

When the order q of $\mathbb{G}$ is not a prime, *e.g.*, a RSA module (*i.e.*, it is the product of two safe primes), RSCDHP may be harder than SCDHP. This is because that even DLP can be solved (hence the SCDHP is also solved), it seems that we still can not solve RSCDHP due to the computation of the quadratic residue modulo a RSA module.

6 Conclusion and Further Works

In this paper, we describe the second short signature scheme from bilinear pairing which is existentially unforgeable under a chosen message attack without using random oracles. The security of our scheme depends on a new complexity assumption we call the k+1 square roots assumption. We discuss the relationship between the k+1 square roots assumption and some related problems and conjectures. Furthermore, the k+1 square roots assumption gives even shorter signatures under the random oracles, a signature is only one element in a finite field.

Another main contribution of this paper is that we first propose some new mathematical problems ($k+1$ RSP, RSCDHP, *etc.*). These problems are not well studied before and we are uncertain their difficulty. For further works, we expect to give a bound on the computational complexity of these problems and seek more their applications for designing cryptographic schemes.

BLS[10], BB04 [5] and ZSS [36] short signature schemes play an important role in many paring-based cryptographic systems. The proposed short signature scheme in this paper is comparable to them and we expect to see many other schemes based on it, such as group signatures [7], aggregate signatures [9] and universal designated-verifier signatures [34].

References

1. M. Abe and T. Okamoto. *A signature scheme with message recovery as secure as discrete logarithm.* Advances in Cryptology -Asiacrypt 1999, LNCS 1716, pp.378-389, Springer-Verlag, 1999.
2. P.S.L.M. Barreto, H.Y. Kim, B.Lynn, and M.Scott, *Efficient algorithms for pairing-based cryptosystems*, Advances in Cryptology-Crypto 2002, LNCS 2442, pp.354-368, Springer-Verlag, 2002.
3. M. Bellare and P. Rogaway, *Random oracles are practical: a paradigm for designing effiient protocols*, First ACM Conference on Computer and Communications Security, pp.62-73, ACM press, 1993.
4. M. Bellare and P. Rogaway, *The exact security of digital signatures - How to sign with RSA and Rabin*, Advances in Cryptology-Eurocrypt 1996, LNCS 1070, pp. 399-416, Springer- Verlag, 1996.
5. D. Boneh and X. Boyen, *Short signatures without random oracles*, Advances in Cryptology-Eurocrypt 2004, LNCS 3027, pp.56-73, Springer-Verlag, 2004.
6. D. Boneh and X. Boyen, *Secure identity based encryption without random oracles*, Advances in CryptologyCrypto 2004, LNCS 3152 , pp. 443C59. Springer-Verlag, 2004.
7. D. Boneh, X. Boyen and H. Shacham, *Short group signatures*, Advances in Cryptology-Crypto 2004, LNCS 3152, pp.41-55, Springer-Verlag, 2004.
8. D. Boneh and M. Franklin, *Identity-based encryption from the Weil pairing*, Advances in Cryptology-Crypto 2001, LNCS 2139, pp.213-229, Springer-Verlag, 2001.
9. D. Boneh, C. Gentry, B. Lynn and H. Shacham, *Aggregate and verifiably encrypted signatures from bilinear maps*, Advances in Cryptology-Eurocrypt 2003, LNCS 2656, pp.272-293, Springer-Verlag, 2003.
10. D. Boneh, B. Lynn, and H. Shacham, *Short signatures from the Weil pairing*, Advances in Cryptology-Asiacrypt 2001, LNCS 2248, pp.514-532, Springer-Verlag, 2001.
11. D. Boneh, I. Mironov and V. Shoup, *A secure signature scheme from bilinear maps*, CT-RSA 2003, LNCS 2612, pp.98-110, Springer-Verlag, 2003.
12. J. Camenisch and A. Lysyanskaya, *A signature scheme with efficient protocols*, SCN 2002, LNCS 2576, pp.274-295, Springer- Verlag, 2003.
13. J. Camenisch and A. Lysyanskaya, *Signature schemes and anonymous credentials from bilinear maps*, CRYPTO 2004, LNCS 3152, pp. 56-72, Springer- Verlag, 2004.
14. J.C. Cha and J.H. Cheon, *An identity-based signature from gap Diffie-Hellman groups*, PKC 2003, LNCS 2567, pp.18-30, Springer-Verlag, 2003.
15. N. Courtois, M. Daum and P. Felke, *On the security of HFE, HFEv- and Quartz*, PKC 2003, LNCS 2567, pp. 337-350. Springer- Verlag, 2003.
16. N.T. Courtois, M. Finiasz and N. Sendrier, *How to achieve a McEliece-based Digital Signature Schem*, Advances in Cryptology-Asiacrypt 2001, LNCS 2248, pp.157-174, Springer-Verlag, 2001.
17. R. Cramer and V. Shoup, *Signature schemes based on the strong RSA assumption*, The Sixth ACM Conference on Computer and Communications Security, pp.46-52, ACM press, 1999.
18. I. M. Duursma and H.-S. Lee, *Tate pairing implementation for hyperelliptic curves* $y^2 = x^p - x + d$, Advances in Cryptology - Asiacrypt 2003, LNCS 2894, pp. 111-123, Springer-Verlag, 2003.
19. M. Fischlin, *The Cramer-Shoup strong-RSA signature scheme revisited*, PKC 2003, LNCS 2567, pp.116-129, Springer-Verlag, 2003.
20. S. D. Galbraith, K. Harrison, and D. Soldera, *Implementing the Tate pairing*, ANTS 2002, LNCS 2369, pp.324-337, Springer-Verlag, 2002.
21. R. Gennaro, S. Halevi and T. Rabin, *Secure hash-and-sign signature without the random oracle*, Advances in Cryptology-Eurocrypt 1999, LNCS 1592, pp.123-139, Springer-Verlag, 1999.
22. S. Goldwasser, S. Micali and R. Rivest, *A 'paradoxical' solution to the signature problem (extended abstract)*, Proc. of FOCS'84, pp. 441-448, 1984.
23. S. Goldwasser, S. Micali and R. Rivest, *A digital signature scheme secure against adaptive chosen-message attacks*, SIAM Journal of computing, 17(2), pp. 281-308, 1988.

24. F. Hess, *Efficient identity based signature schemes based on pairings*, SAC 2002, LNCS 2595, pp.310-324, Springer-Verlag, 2002.
25. H. Krawczyk and T. Rabin, *Chameleon hashing and signatures*, Proc. of NDSS 2000, pp.143-154, 2000.
26. U. Maurer, *Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms*, Advances in Cryptology-Crypto 94, LNCS 839, pp.271-281, Springer-Verlag, 1994.
27. D. Naccache and J. Stern, *Signing on a postcard*, Financial Cryptography 2000, LNCS 1962, pp. 121-135, Springer-Verlag, Berlin, 2000.
28. K. Nyberg and R. Rueppel, *A new signature scheme based on the DSA, giving message recovery*, Proceedings of the 1st ACM conference on communications and computer security, pp. 58-61, 1993.
29. J. Patarin, N. Courtois and L. Goubin, *QUARTZ, 128-bit long digital signatures*, CT-RSA 2001, LNCS 2020, pp. 282-297, Springer-Verlag, 2001.
30. K.G. Paterson, *ID-based signatures from pairings on elliptic curves*, Electron. Lett., Vol.38, No.18, pp.1025-1026, 2002.
31. A.R. Sadeghi and M. Steiner,*Assumptions related to discrete logarithms: why subtleties make a real difference*, Advances in Cryptology-Eurocrypt 2001, LNCS 2045, pp.243-260, Springer-Verlag, 2001.
32. A. Shamir, *Identity-based cryptosystems and signature schemes*, Advances in Cryptology-Crypto 1984, LNCS 196, pp.47-53, Springer-Verlag, 1984.
33. A. Shamir and Y. Tauman, *Improved online/offine signature schemes*, Advances in Cryptology-Crypto 2001, LNCS 2139, pp. 355-367, Springer-Verlag, 2001.
34. R. Steinfeld, L. Bull, H. Wang, and J. Pieprzyk, *Universal designated-verifier signatures*, Advances in Cryptology-Asiacrypt 2003, LNCS 2894, pp.523-542, Springer-Verlag, 2003.
35. R. Zhang, J. Furukawa and H. Imai, *Short signature and universal designated verifier signature without random oracles*, ACNS 2005, LNCS 3531, pp. 483C498, Springer-Verlag, 2005.
36. F. Zhang, R. Safavi-Naini and W. Susilo, *An efficient signature scheme from bilinear pairings and its applications*, PKC 2004, LNCS 2947, pp.277-290, Springer-Verlag, 2004.