A Short Signature Scheme in the Standard Model

Li Kang, Xiaohu Tang, Xianhui Lu, Jia Fan
Email:kangli@mars.swjtu.edu.cn

Lab. of Information Security & National Computing Grid, SWJTU, Chengdu, China

Abstract. In this paper, by elaborately choosing the parameters of Waters Hash function, we propose a new efficient signature scheme. It is shown that the scheme is secure against strongly unforgeable chosen-message attacks in the standard model under Computational Diffie-Hellman (CDH) assumption. Further, among all the known secure signatures in the standard model, our scheme is the shortest one and has the efficient security reduction as well.

Key Words: Short signature, Strongly unforgeable, CDH

1 Introduction

The design of an efficient and secure signature scheme is one of the focus of interest in cryptography. Due to the rigorous pressure of the bandwidth, short signatures are most favorable with respect to the efficiency. On the other hand in view of the security, strongly unforgeable signatures, which ensure the adversary cannot even produce a new signature for any previously signed message, are very desirable too. But up to now, in the standard model all the existing signature schemes do not satisfy the two requirements simultaneously.

In 2004, Boneh and Boyen (BB) constructed a short and strongly unforgeable signature scheme in the standard model [1]. However, its security reduces to q-strong Diffie-Hellman (q-SDH) assumption, which is a stronger assumption compared with the standard computational Diffie-Hellman (CDH) assumption. The signature needs one element in pairing group $\mathbb{G}_p$ and one element in $\mathbb{Z}_p$ based on q-SDH assumption, where p is a prime determined by the security level. Later, several researchers analyzed the security of strong Diffie-Hellman assumption [7, 5]. They pointed out that it has computational complexity reduced by $O(\sqrt{q})$ from that of the discrete logarithm problem. Hence, they recommend that any scheme based on the q-SDH assumption should increase the size of the elements in groups, e.g. the elements in $\mathbb{G}_p$ and $\mathbb{Z}_p$, (by up to 50% more bits) for any given security level, in contrast to the elements in groups built on the CDH assumption.

In 2005, Waters proposed an efficient identity-based (IBE) encryption scheme based on CDH assumption [12]. The key technique to Waters IBE scheme is the usage of the so called Waters Hash function. In [12], Waters also gave a signature scheme by the Waters Hash function. The signature needs two elements in pairing group $\mathbb{G}_p$, but it is not strongly unforgeable.

In 2006, Boneh, Shen and Waters (BSW) presented a strongly unforgeable signature from Waters signature scheme [4]. The signature needs two elements in pairing group $\mathbb{G}_p$ and one element in $\mathbb{Z}_p$ based on CDH assumption.

Our Contributions. In this work we construct a short and strongly unforgeable signature scheme based on the CDH assumption in the standard model. The signature scheme is simple and efficient. It needs only one element in pairing group $\mathbb{G}_p$ and one element in $\mathbb{Z}_p$. In generally for a given security level, the size of the elements in $\mathbb{G}_p$ is much bigger than one in $\mathbb{Z}_p$ (c.f. Table 2 in Section 4). Therefore, our scheme has the shortest size amongst all the known signatures in the standard model.

The form of our signature is similar to BLS signature [3], which is a short signature in the random oracle model. To achieve our signature in the standard model, we employ the Waters Hash function [12]. In particularly, we elaborately select the parameters of Waters Hash function. This selection enables us to construct a short and strongly unforgeable signature with tight security reduction based on the CDH assumption in the standard model.

2 Preliminaries

2.1 The Target Collision Resistant Hash Function

The notion of target collision resistant TCR family of hash functions was presented by Cramer and Shoup [8]. It is a special case of universal one-way hash function (UOWH) family introduced by Naor and Yung [11], where a UOWH family can be built from arbitrary one-way functions [11].

In a TCR family, given a randomly chosen tuple of group elements x and a randomly chosen hash function H, it is infeasible for an adversary $\mathcal{A}$ to find a $y \neq x$ such that $H(x) = H(y)$. In practice, one can use a dedicated cryptographic hash function, like SHA-1 or SHA-256 for 80-bit or 128-bit security levels, respectively. Let n be the output length of the hash function H determined by the security parameter k. For an efficient adversary $\mathcal{A}$, we define

$$\mathbf{Adv}_{\mathcal{TCR}}^{hash-tcr}(k) = Pr[\mathcal{A}\ succeeds].$$

Hash function is said to be target collision resistant if the advantage function $\mathbf{Adv}_{\mathcal{TCR}}^{hash-tcr}(k)$ is a negligible function in k for all polynomial-time adversaries $\mathcal{A}^{tcr}$.

2.2 The Computational Diffie-Hellman (CDH) Assumption

Let $\mathbb{G}$ be a group of prime order p, whose size is determined by the security parameter k. The Computational Diffie-Hellman assumption (CDH) supposes that given the input (g, g^a, g^b), where g is a generator of a group $\mathbb{G}$, it should be computationally infeasible to compute g^{ab}. More precisely, the Computational Diffie-Hellman assumption is said to be secure if the advantage function $\mathbf{Adv}_{\mathcal{A}}^{cdh}(k)$ is a negligible function in k for all polynomial-time adversaries $\mathcal{A}^{cdh}$, where

$$\mathbf{Adv}_{\mathcal{A}}^{cdh}(k) = Pr[\mathcal{A}(g, g^a, g^b) \rightarrow g^{ab}].$$

2.3 Bilinear Groups

Let $\mathbb{G}$ and $\mathbb{G}_1$ be a pair of group of prime order p. Let g be a generator of $\mathbb{G}$. A bilinear pairing is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ with two properties:

1. bilinearity: $e(g^a, g^b) = e(g, g)^{ab}$, $\forall a, b \in \mathbb{Z}_p$;
2. non-degeneracy: $e(g, g) \neq 1$.

We say that $\mathbb{G}$ is a bilinear group if the group operation in $\mathbb{G}$ can be computed efficiently, and there exists a group $\mathbb{G}_1$ and an efficiently computable bilinear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ as above.

3 Signature and Security Model

A signature is a tuple (**Setup**, **Sign**, **Verify**) where

Setup. Inputs a security parameter k, outputs public key and secret key *pk*-*sk* pair (pk, sk).

Sign. Inputs a message M, public key pk and secret key sk, outputs a signature σ.

Verify. Inputs a message M, signature σ and public key pk, outputs 1 or 0 for valid or invalid.

Strong existential unforgeability under an adaptive chosen-message attack(sucma) is defined using the following game:

Setup. The challenger gives the adversary the public key pk and keeps the private key sk to itself.

Signature Queries. The adversary issues signature queries $M_1, \ldots, M_q$. To each query M_i the challenger responds by running **Sign** to generate a signature σ_i of M_i and sending σ_i to the adversary. These queries may be asked adaptively so that each query M_i may depend on the replies to $M_1, \ldots, M_{i-1}$.

Output. Finally the adversary outputs a pair (M^*, σ^*). The adversary wins if σ^* is a valid signature of M^* according to **Verify** and (M^*, σ^*) is not among the pairs (M_i, σ_i) generated during the query phase.

We define the advantage of an adversary $\mathcal{A}$ in attacking the signature scheme as the probability that $\mathcal{A}$ wins the above game, taken over the random bits of the challenger and the adversary.

Definition 1 *A signature scheme is $(t, q, \varepsilon_{\mathcal{A}})$-strongly existentially unforgeable under an adaptive chosen-message attack(sucma) if no t-time adversary $\mathcal{A}$ making at most q signature queries has advantage at least $\varepsilon_{\mathcal{A}} = \boldsymbol{Adv}_{\mathcal{A}}^{sucma}$ in the above game.*

4 Signature Scheme and Security Proof

4.1 Signature Scheme

Setup. Let $\mathbb{G}$ be a bilinear group of prime order p, where security parameter k determines the size of $\mathbb{G}$. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_1$ be the bilinear map. Firstly we choose two target collision-resistant hash functions $H_1 : \{0,1\}^* \rightarrow \{0,1\}^n$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_p$, where the integer n is determined by the security parameter k. Next we pick a random generator $g \in \mathbb{G}$, choose random $a \in \mathbb{Z}_p$ and $h \in \mathbb{G}$, n-length vector $\overrightarrow{u} = (u_i)$, whose elements are chosen at random from $\mathbb{G}$. Finally, the public key pk and secret key sk are given by

$$pk = (g, g^a, h, H_1, H_2, \overrightarrow{u} = (u_i)), \quad sk = (a).$$

Sign. To sign a message M, the signer chooses a random $r \in \mathbb{Z}_p$, computes $v = H_1(M, r)$ and gives the signature as

$$(\sigma_1, \sigma_2) = ((h^{H_2(M,r)} \prod_{i \in \mathcal{V}} u_i)^a, r)$$

where the set $\mathcal{V}$ is formed by all the is that the ith bit v_i of v is 1.

Verify. To verify a signature (σ_1, σ_2) on M, the receiver computes $v = H_1(M, \sigma_2)$ and tests

$$e(\sigma_1, g) = e((h^{H_2(M,\sigma_2)} \prod_{i \in \mathcal{V}} u_i), g^a) \tag{1}$$

if it holds, accepts the signature, otherwise rejects it.

4.2 Security Proof

Theorem 1 *If an adversary $\mathcal{A}$ can forge a valid signature with advantage $\varepsilon_{\mathcal{A}} = \boldsymbol{Adv}_{\mathcal{A}}^{sucma}$ and running time $\boldsymbol{Time}_{\mathcal{A}}(k)$ we construct a challenger $\mathcal{B}$ breaking the CDH assumption with advantage $\varepsilon_{\mathcal{B}} = \boldsymbol{Adv}_{\mathcal{B}}^{cdh}(k)$ and running time $\boldsymbol{Time}_{\mathcal{B}}(k)$ with*

$$\varepsilon_{\mathcal{B}} \geq \frac{\varepsilon_{\mathcal{A}}}{2}(1 - \boldsymbol{Adv}_{\mathcal{TCR}}^{hash-tcr}(k)),$$

$$\boldsymbol{Time}_{\mathcal{B}}(k) \leq \boldsymbol{Time}_{\mathcal{A}}(k) + qt_s,$$

where q is an upper bound on the number of signature queries made by adversary $\mathcal{A}$. The t_s denotes the time required for one time signature query computation.

***Proof*:** In this proof the challenger $\mathcal{B}$ will solve the CDH assumption successfully with the help of the forge ability of the adversary $\mathcal{A}$, who interacts with $\mathcal{A}$ as follows.

Setup. Assume that the integer n determined by the security parameter k is a multipler of 4. The challenger $\mathcal{B}$ obtains the (g, g^a, g^b) from CDH assumption. It picks n-length vector $\overrightarrow{x} = (x_i)$ and $\overrightarrow{y} = (y_i)$, whose elements are chosen at random from $\mathbb{Z}_p^*$ and the (x_i) satisfy parity distributing uniformity. The challenger sets $u_i = (g^b)^{(-1)^{x_i}} g^{y_i}$ and $\overrightarrow{u} = (u_i)$. It selects a random $z \in \mathbb{Z}_p^*$, sets $h = g^z$. Finally, it chooses two target collision-resistant hash functions $H_1 : \{0,1\}^* \rightarrow \{0,1\}^n$ and

$H_2 : \{0,1\}^* \to \mathbb{Z}_p$. For ease of analysis we define two functions: $F(v) = \sum_{i\in\mathcal{V}}(-1)^{x_i} \pmod p$ and $J(v) = \sum_{i\in\mathcal{V}} y_i \pmod p$. The public key pk and secret key sk are given by

$$pk = (g, g^a, h, H_1, H_2, \overrightarrow{u} = (u_i)), \quad sk = (a(unknown), \overrightarrow{x} = (x_i), \overrightarrow{y} = (y_i), z).$$

Signature queries. The challenger $\mathcal{B}$ receives a signature query on M_j. It selects a random $r \in \mathbb{Z}_p$, which satisfies $F(v) = 0 \pmod p$ for $v = H_1(M_j, r)$, and gives the signature as

$$(\sigma_{1,j}, \sigma_{2,j}) = ((h^{H_2(M_j,r)} \prod_{i\in\mathcal{V}} u_i)^a, r) = (((g^z)^{H_2(M_j,r)}(g^b)^{F(v)} g^{J(v)})^a, r) = ((g^a)^{J(v)+zH_2(M_j,r)}, r).$$

It is obvious that $(\sigma_{1,j}, \sigma_{2,j})$ is a valid signature on M_j. Later, we will determine the average selection times of such integers r for an given message M_j.

Output. At this phase the adversary $\mathcal{A}$ returns a signature $(\sigma_1^*, \sigma_2^*) = ((h^{H_2(M^*,r^*)} \prod_{i\in\mathcal{V}} u_i)^a, r^*)$ on any M^*. The challenger firstly verifies it by (1), if not holds rejects the signature. Next computes $v^* = H_1(M^*, r^*)$ and tests $F(v^*) = 0$, if holds aborts (we will compute the abort probability in the forthcoming discussions), otherwise the challenger can computes g^{ab} as

$$g^{ab} = (\frac{\sigma_1^*}{(g^a)^{J(v^*)+zH_2(M^*,\sigma_2^*)}})^{\frac{1}{F(v^*)}} = (\frac{((g^b)^{F(v^*)} g^{J(v^*)+zH_2(M^*,\sigma_2^*)})^a}{(g^a)^{J(v^*)+zH_2(M^*,\sigma_2^*)}})^{\frac{1}{F(v^*)}}.$$

In **Signature queries** phase, the challenger is able to select a certain r with $F(v) = 0$, where $v = H_1(M_j, r)$. The ability is guaranteed by the following reasonable selections:

- The challenger selects a target collision-resistant hash function H_1 of output length n, such that H_1 outputs all 0 and all 1 with negligible probability.
- The challenger chooses $\overrightarrow{x} = (x_i)$, $i = (1, \ldots, n)$, half odd number and half even.
- The challenger sets $u_i = (g^b)^{(-1)^{x_i}} g^{y_i}$, such that the exponents of g^b that assemble a sequence length of $n$bits, which comprises by 1 and -1 randomly.

Denote Pr_{even} the possibility of v with the even numbers of 1. Suppose that v has $2m$ bits 1 with the possibility Pr_m, i.e., $Pr_{even} = \sum_{m=0}^{n/2} Pr_m$. From Probability Theory, we have

$$Pr_m[F(v) = 0] = \frac{C_{n/2}^m C_{n/2}^m}{C_n^{2m}}.$$

Define $\mathcal{C}_m = (C_{n/2}^m)^2 / C_n^{2m}$. It is easy to verify that $\mathcal{C}_m = \mathcal{C}_{n/2-m}$, and $\mathcal{C}_m > \mathcal{C}_{m+1}$, $0 \le m < n/4$. Hence, with m ranging from 1 to $n/2$, the maximum $\mathcal{C}_{\max}$ and the minimum $\mathcal{C}_{\min}$ are respectively

$$C_{\max} = \mathcal{C}_1 = \frac{(C_{n/2}^1)^2}{C_n^2} = \frac{n/2}{n-1} \approx \frac{1}{2},$$

$$\mathcal{C}_{\min} = C_{n/4} = \frac{(C_{n/2}^{n/4})^2}{C_n^{n/2}}.$$

In the following, we give the corresponding minimum $\mathcal{C}_{\min}$ for the security levels 80-bit, 128-bit, 256-bit and 512-bit, which are corresponding for the output length n of the secure hash functions, i.e., $n = 160, 256, 512, 1024$(bits), respectively [9].

Table 1 n and $\mathcal{C}_{\min}$

security level	n	$\mathcal{C}_{\min}$
80	160	0.125567
128	256	0.0994438
256	512	0.0704205
512	1024	0.0498313

In addition for the above given security levels, Table 2 derived from [5] is given as follows to illustrate the elements size representation of pairing group $\mathbb{G}_p$ and $\mathbb{Z}_p$ under the CDH assumption. Herein we consider concrete case of pairings on supersingular (SS) curves of large characteristic used in [2], and pairings on MNT curves [10] as described in [3].

Table 2 Size Representation(bits)

	SS	MNT	MNT
	80-bit security	80-bit security	128-bit security
$\mathbb{Z}_p$	160	160	256
$\mathbb{G}$	512	171	512

From Table 2, we notice that the elements in $\mathbb{Z}_p$ is $n$bits at the the security level $n/2$. Thus for the integer $r \in \mathbb{Z}_p$, the size $|r| = nbits$ is as the same as the output length of the hash function H_1. Then when the challenger $\mathcal{B}$ signs by selecting r uniformly from $\mathbb{Z}_p$, it is reasonable to assume that $Pr_{even} = \sum_{m=0}^{n/2} Pr_m = 1/2$.

Consequently for the signature signed by $\mathcal{B}$,

$$\begin{aligned} Pr[F(v) = 0] &= \sum_{m=0}^{n/2} Pr_m \cdot Pr_m[F(v) = 0] \\ &\geq C_{\min} \sum_{m=0}^{n/2} Pr_m \\ &= C_{\min} Pr_{even} \\ &= \frac{C_{\min}}{2}. \end{aligned}$$

Generally speaking, at least the signature scheme is required to have 80-bit security level, i.e., n=160bits. Accordingly, $Pr[F(v) = 0] \geq 0.06278$, averagely if selecting 16 times random r, the challenger will get a $v = H_1(M_j, r)$ such that $F(v) = 0 \,(\bmod\, p)$. If the security level is extended to 128-bit, 256-bit, or 512-bit, then averagely the challenger obtains a certain r satisfying $F(v) = 0$ from at most 21, 29, or 40 times selections.

In the **output** phase, the challenger will abort in two independent cases:

1. The adversary $\mathcal{A}$ outputs a forge signature with $F(v^*) = 0\,(\bmod\, p)$. Obviously $F(v^*) = 0\,(\bmod\, p)$ only occurs for $v^* = H_1(M^*, r^*)$ having even number of 1. Similar to the analysis in **Signature Query** phase,

$$\begin{aligned}
Pr[abort]_{F(v^*)=0} &= \sum_{m=0}^{n/2} Pr_m \cdot Pr_m[F(v^*) = 0] \\
&= \sum_{m=1}^{n/2-1} Pr_m \cdot Pr_m[F(v^*) = 0] \\
&\le C_{\max} \sum_{m=1}^{n/2-1} Pr_m[F(v^*) = 0] \\
&= C_{\max} Pr_{even} \\
&= \frac{Pr_{even}}{2},
\end{aligned}$$

where the second identity comes from the fact that the Hash function H_1 outputs $m = 0$ or $n/2$ with negligible probability, and Pr_{even} is the possibility of v^* submitted by $\mathcal{A}$ with the even numbers of 1.
In **Signature Query** phase, all the signatures on M_j, signed by the challenger $\mathcal{B}$, satisfies the property that $v = H_1(M_j, r)$ possesses even number of 1. In **Output** phase, a clever adversary $\mathcal{A}$ may observe this property. So, it is most likely that $\mathcal{A}$ always generates forge signatures with $v^* = H_1(M^*, r^*)$ having even number of 1, i.e., $Pr_{even} = 1$. Then,

$$Pr[abort]_{F(v^*)=0} \le \frac{1}{2},$$

2. The adversary $\mathcal{A}$ submit a tuples $(M^*, r^*) \ne (M_j, r)$ to make $H_1(M^*, r^*) = H_1(M_j, r)$ or $H_2(M^*, r^*) = H_2(M_j, r)$. This case occurs with probability

$$Pr[abort]_{target\ hash\ collision} = \mathbf{Adv}_{\mathcal{TCR}}^{hash-tcr}(k).$$

That is, the simulation will not abort with probability

$$\begin{aligned}
Pr[\overline{abort}] &= (1 - Pr[abort]_{F(v^*)=0})(1 - Pr[abort]_{target\ hash\ collision}) \\
&\ge \frac{1}{2}(1 - \mathbf{Adv}_{\mathcal{TCR}}^{hash-tcr}(k)).
\end{aligned}$$

Thus, we have

$$\varepsilon_{\mathcal{B}} \ge \frac{\varepsilon_{\mathcal{A}}}{2}(1 - \mathbf{Adv}_{\mathcal{TCR}}^{hash-tcr}(k)), \quad \mathbf{Time}_{\mathcal{B}}(k) \ \le\ \mathbf{Time}_{\mathcal{A}}(k) + qt_s.$$

In particular, we should emphasize that the reduction probability is always $\varepsilon_{\mathcal{B}} \ge \frac{\varepsilon_{\mathcal{A}}}{2}(1 - \mathbf{Adv}_{\mathcal{TCR}}^{hash-tcr}(k))$, whatever n = 160, 256, 512, 1024 etc. That is, our scheme possess an advantage that the security reduction will maintain the same efficiency with the security level enhances (the n increase).

4.3 Similarity to Waters hash function

Indeed, our proof is based on a modification of Waters Hash function [12], by simply replacing $u_i = (g^b)^{x_i} g^{y_i}$ in Waters scheme with $u_i = (g^b)^{(-1)^{x_i}} g^{y_i}$. Whereas the slight modification is powerful to ensure that our signature has an efficient security reduction.

5 Efficiency Comparison and Conclusion

Table 3 gives the comprehensive comparison among our signature scheme and other three schemes. In particular, the size of elements in $\mathbb{G}$ and $\mathbb{Z}_p$ are based on CDH assumption and the size of elements in $\mathbb{G}'$ and $\mathbb{Z}'_p$ are based on q-SDH assumption, denotes by $|\mathbb{G}|, |\mathbb{Z}_p|, |\mathbb{G}'|$ and $|\mathbb{Z}'_p|$, respectively. As mentioned in Section 1, it should be noticed that at the same security level $|\mathbb{G}'|$ and $|\mathbb{Z}'_p|$ should increase by up to 50% (bits) on the basis of $|\mathbb{G}|$ and $|\mathbb{Z}_p|$, respectively, to compensate for Cheon's attack [7].

Table 3 Comprehensive Comparison

Scheme	Signature size	Standard Model	Assumption	Security Reduction	Strong unforgeable
Ours	$\|\mathbb{G}\| + \|\mathbb{Z}_p\|$	$\surd$	CDH	1	$\surd$
BB [1]	$\|\mathbb{G}'\| + \|\mathbb{Z}'_p\|$	$\surd$	$q\text{-}SDH$	1	$\surd$
BSW [4]	$2\|\mathbb{G}\| + \|\mathbb{Z}_p\|$	$\surd$	CDH	$\frac{1}{nq}$	$\surd$
Waters [12]	$2\|\mathbb{G}\|$	$\surd$	CDH	$\frac{1}{nq}$	–

Based on Tables 2 and 3, the size of the various signatures is then given in Table 4. From it, one can see that our signature is always the shortest no matter what security levels and what pairing groups are used.

Table 4 Size Comparison(bits)

Scheme	SS	MNT	MNT
	80-bit security	80-bit security	128-bit security
Ours	672	331	768
BB [1]	1008	497	1152
BSW [4]	1184	502	1280
Waters [12]	1024	342	1024

Therefore, by slightly modifying the parameters selection of Waters Hash function, we construct a short and strongly unforgeable signature scheme based on the standard computational Diffie-Hellman(CDH) assumption in the standard model. The signature scheme is simple and has tight security reduction. It needs only one element in pairing group $\mathbb{G}_p$ and one element in $\mathbb{Z}_p$.

References

1. D. Boneh and X. Boyen, “Short signatures without random oracles,” *in Advances in Cryptology - Eurocrypt 2004, Lecture Notes in Computer Science*, vol. 3027, Springer-Verlag, pp. 56-73, 2004.
2. D. Boneh and M. K. Franklin, “Identity-based encryption from the weil pairing,” *SIAM Journal on Computing*, vol. 32(3), pp. 586-615, 2003.
3. D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the Weil pairing,” *J. of Cryptology*, 17(4), pp: 297-319, 2004.
4. D. Boneh, E. Shen and B. Waters, “Strongly unforgeable signatures based on computational diffie-hellman,” *in Proc. of PKC 2006,Lecture Notes in Computer Science*, vol. 3958, Springer-Verlag, pp. 229-240, 2006.
5. X. Boyen, “The BB1 Identity-Based Cryptosystem : A Standard for Encryption and Key Encapsulation,” IEEE Standard draft, P1363.3, available at:http://grouper.ieee.org/groups/1363/email/discuss/msg00116.html.
6. R. Canetti, S. Halevi, and J. Katz, “Chosen-ciphertext security from identity-based encryption,” *in Advances in Cryptology - Eurocrypt 2004, Lecture Notes in Computer Science*, vol. 3027, Springer-Verlag, pp. 207-222, 2004.
7. J. H. Cheon, “Security analysis of the Strong Diffie-Hellman problem,” *in Advances in Cryptology - Eurocrypt 2006, Lecture Notes in Computer Science*, vol. 4004, Springer-Verlag, pp. 1-11, 2006.
8. R. Cramer and V. Shoup, “Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack,” *SIAM Journal on Computing*, Vol. 33, Number 1, pp. 167-226, 2003.
9. FIPS PUB 180-2, “Secure hash standard,” available at http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf
10. A. Miyaji, M. Nakabayashi, and S. Takano, “New explicit conditions of elliptic curve traces for FR-reduction,” *IEICE Trans. Fundamentals*, E84-A(5), pp. 1234-1243, 2001.
11. M.Naor and M.Yung, “Universal One-Way Hash Functions and their Cryptographic Applications,” STOC 1989, pp. 33-43, 1989.
12. B. Waters. “Efficient identity-based encryption without random oracles,” *in Advances in Cryptology - Eurocrypt 2005, Lecture Notes in Computer Science*, vol. 3494, Springer-Verlag, pp. 114-127, 2005.