A Stamped Hidden-signature Scheme Utilizing The Elliptic Curve Discrete Logarithm Problem

Mohamed M. Rasslan

Electrical and Computer Engineering Department, Concordia University, Montreal, Canada
1455 De Maisonneuve Blvd. West, Montreal, Quebec, Canada H3G 1M8 (Email: m_rassla@encs.concordia.ca)
(Received Aug. 28, 2009; revised and accepted Dec. 31, 2009)

Abstract

Based on the anonymity that digital signatures provide to users and messages, digital signatures can be classified as hidden, weak, interactive, or strong blind signatures. The hidden blind signature hides the signed message from the signer's vision during his interaction with an honest requester. Later on, after revealing the message the signer can easily link the message-signature pair. The hidden blind signature application deals with message anonymity only and cannot be done through a strong blind signature; the notary service is one example of a hidden blind signature. In this paper we propose a hidden blind signature scheme that utilizes bilinear pairing over elliptic curves. The proposed scheme requires smaller key sizes for the same level of security compared to schemes not utilizing bilinear pairings. The proposed scheme allows the signer to add information in the signed message. The requester cannot modify either this information or the signed message. This added information stamps the signature with a certain date and place which we see as an essential requirement in applications such as notary service (testament application) and patent time proof. In notary service, there is no conflict of interest between the signer and the requester of the signature. There is no need to have a trusted party to authenticate the temporal or spatial information. Instead, the signature requester will embed this information into the message body which is hidden from the signer. After issuing the signature by the signer, the requester can verify that the signature has the designated date and place. This is under the assumption that the signer has to perform the signing process on the same day and that she is free to sign at any time that day. This date-stamping is very important in case the signers' signature key is stolen or compromised. The proposed scheme is proved to be secure against an existential adaptive chosen message attack.

[0]A preliminary version of this paper was presented at the first international workshop on Communications Security & Information Assurance (CSIA), Ankara, Turkey, 2010.

1 Introduction

In traditional paper-based signing, the validation of documents can be ensured as long as they are kept in sealed envelopes. The concept of blind signatures, as first introduced by Chaum [8, 9], was a breakthrough in achieving the digitalization of services. It is a digital signature form that maps paper-based systems to an electronic version that lends anonymity to the message (blindness) and gives anonymity to the user (unlinkability). Blind signatures are the core of applications such as e-coins and e-voting. Many blind signatures that satisfy anonymity and unlinkability have been proposed [3, 31, 33]. Blind signatures are publicly verified by any third party and meet the requirements of privacy-oriented protocols that have a conflict of interest between the signer and message's author as in electronic money and electronic election protocols.

Chaum's scheme has three active parties: the message's author, who requests the signature on the message, the signer, and the verifier who ensures that the signature does belong to the signer. The scheme starts with the requester (author) blinding the message then sends this blind message to a singer who signs it and sends the signed blind message back to the requester. Then, the requester unblinds the message. The result is a digital signature of the original message. As we see, a blind signature is composed of four ordered algorithms: blinding, signing, unblinding, and verifying. In Chaum's scheme the resultant message-signature pair is unlinkable and is classified as a strong blind signature in [15, 16, 17]. In the unblinding algorithm, the requester modifies the signature and produces version that is unlinkable to the one generated by the signer. Hence, the signer has no way to resume this link again even after revealing the message. Some applicators find this unlinkability undesirable but at the same time, they are interested in blinding messages. Our hidden blind signature scheme enables the signer to insert a piece of temporal or spatial information in the signature and prevents the requester from modifying either this piece of information or the signature. All this is done without disclosing the message. Horster *et al.*'s classifi-

cation in [16, 17] defines hidden signatures into message hidden signatures and s-hidden signatures where s is the signature parameter. Similarly, our protocol is a message hidden signature protocol. In short, we use "hidden signature" instead of "message hidden signature." At the abstract level, hidden signatures reform the terminology of blind signature so that it blinds the signer during the signing process, not the signature.

Horster *et al.* [15, 16, 17] classified the blind signature into four classes depending on the anonymity strength given by the signature: the hidden, the weak blind, the interactive and the strong blind signatures.

In hidden blind signatures, the signer does not know the message to be signed but he knows the signature parameters. The signer has the chance to store these parameters and can recognize the signature later by comparing them with a given signature.

In weak blind signatures, the signer does not know either the message to be signed or the signature parameters. But, after revealing the message, he can easily link his signature to that message. The reason behind the signer's ability to recognize the message-signature pair is the existence of a relationship between the blinded signature parameters and the unblinded parameters.

Interactive blind signatures are similar to weak blind signatures in their generation, except that interactive blind signatures use interactive proof to demonstrate the knowledge of a signature, so the signer does not have the ability to link the blinded and the unblinded signature parameters. Hence, the signer cannot link a given message to his stored parameters.

Finally, the strong blind signature grants full anonymity and the signer cannot link his stored parameters and the public signature parameters. The strong blind signature causes a risk of misuse, for example in money laundering, blackmailing or asking ransom.

This paper proposes a hidden signature scheme. The proposed scheme is motivated by the work of [16, 7] and it utilizes the bilinear pairing cryptosystems from the Gap-Diffie-Hellman groups.

The organization of the rest of the paper is as follows: the next section is a brief survey about related works and bilinear pairing cryptosystems. In Section 3, some computational preliminaries are presented. Section 4 presents a description of the proposed scheme. An analysis of the scheme is presented in Section 5.

2 Related Work

Camenisch *et al.* [7] proposed the first blind signatures based on the discrete logarithm problem. Subsequently, in 1995, Harn [18] proved that unlinkability (untraceability) cannot be achieved in Camenisch *et al.*'s scheme. But, according to Horster *et al.* [15, 17], Camenisch *et al.*'s scheme could be classified as a hidden signature. Also, Horster *et al.* in [16, 17] introduced a hidden signature protocol that is based on the ideas of the testimonial scheme [19] and the Meta-ElGamal signature scheme [20]. The proposed hidden signature scheme, in this paper, utilizes bilinear pairing. Since Joux's work [21], bilinear pairings, especially modified Weil and Tate pairings, have been a practical and efficient means for cryptographic protocols [4, 5]. This attracted cryptographers to use bilinear pairings in public key cryptography and exploded the field of pairing-based cryptography over the past few years. Basically, bilinear pairing is a mapping between two functional cryptographic groups which allows for new cryptographic schemes to take place. The resultant scheme is based on problem reduction in one group to a different easier problem in the other group. In the literature, the first group is called a Gap Group. In a Gap Group, the Computational Diffie-Hellman problem is hard and the Decisional Diffie-Hellman problem is easy. The word easy means that, the pairing reduces the Decisional Diffie-Hellman problem to an easy problem in the second group. On the other hand, the Computational Diffie-Hellman problem remains hard [6]. Understanding the Weil and Tate pairings requires complex mathematics. Fortunately, cryptographers can deal with it abstractly, through group structure and mapping properties. The literature proposes many interesting schemes based purely on abstract bilinear maps. In the next section we introduce the mathematical background of bilinear maps.

3 Preliminary

Bilinear maps are the tool of pairing based cryptography, which is a hot topic that started with an identity-based encryption scheme by Boneh and Franklin in 2001 [4]. An abstract understanding of this tool requires knowledge of Gap Diffie-Hellman groups and bilinear groups. Gap Diffie-Hellman groups created from disjointing Computational and Decisional Diffie-Hellman problems. Bilinear groups are based on the existence of a bilinear map. Simply, a bilinear map is a function with certain properties. Let G be an additive cyclic group of prime order p, and P is its generator. In this group, the well-known Diffie-Hellman problems carry on as follows [4, 5, 27].

3.1 Diffie-Hellman Problems

Computational Diffie-Hellman (CDH). Given P, aP, $Q \in G$, compute $aQ \in G$.

Decisional Diffie-Hellman (DDH). Given P, aP, Q, $bQ \in G$, decide whether a equals b. Quadruples of this form (P, aP, Q, bQ) are named Diffie-Hellman quadruples.

Inverse Computational Diffie-Hellman problem (InvCDH). Given P, xP, outputs $\frac{1}{x}P$. The value $\frac{1}{x}$ is the multiplicative inverse of $x \in Z_p^*$.

Gap Diffie-Hellman Groups (GDH) are examples

of gap problems presented in [26]. There are many subgroups of group Z_q^* that have prime orders, and both the CDH and DDH assumptions are believed to be held. The subgroup G with the prime order p is one of these. However, on certain elliptic-curve groups, the DDH problem is easy to solve, whereas CDH is believed to be hard [5]. Such groups are named Gap Diffie-Hellman (GDH) groups. Hence, if G belongs to these specific elliptic-curve groups, we call it a Gap Diffie-Hellman group.

3.2 Elliptic Curve Cryptosystems

The theory of elliptic curve has been intensively studied in the pure mathematics field for 160 years and it was applied for factoring large integers in early 1980's. In 1984, Miller and Koblitz independently found that the group of points on an elliptic curve is a proper group on which the discrete logarithm is intractable. These groups are suitable for implementing El-Gamal type cryptosystems by replacing the residue group of integers in El-Gamal by these groups.

Elliptic Curve Groups. If F_q is a field and E is an elliptic curve then $E(F_q)$ is a group. We read $E(F_q)$, elliptic curve E over field F_q which indicates the set of points on E along with only one operation (addition) defined for $E(F_q)$. So it is impossible to multiply or divide elements of $E(F_q)$ [24]. Multiplication of a point P on an elliptic curve by an integer n is the result of adding a point to itself n times [24].

3.3 Bilinear Maps

Bilinear Groups. Until now, there have not been known any implementable example of GDH groups except bilinear maps, which have an additional structure. A bilinear group is any group that possesses such a map e, and on which CDH is hard.

Bilinear Maps. Assume that G is an additive group and G_T is a multiplicative group such that $|G| = |G_T| = |p|$, where p is a prime number. P is the generator of G. Then, the map $e : G \times G \rightarrow G_T$ is a computable bilinear map if it satisfies:

1) Computability: There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G$.
2) Bilinearity: for all $P, Q \in G$ and $a, b \in Z$, we have $e(aP, bQ) = e(P, Q)^{ab}$.
3) Non-Degeneracy: $e(P, P) \neq 1$. In other words, if P is a generator of G, then $e(P, P)$ generates G_T.

The work of Joux and Nguyen in [22] completed the bilinear maps story by overcoming the last obstacle preventing it from beeing a practical and efficient example of GDH. They illustrated that a bilinear map e provides an algorithm for solving DDH. For a tuple (P, aP, Q, bQ) we have $a = b \; modp \Leftrightarrow e(Q, aP) = e(bQ, P)$. As a result, if a group G is a bilinear group then G is also a GDH group. (The opposite does not have to be true.)

Bilinear Diffie-Hellman Problem. The group G is a subgroup of the additive group of points of an elliptic curve$E(F_q)$. The group G_T is a subgroup of the multiplicative group of finite field F_q^* and $|G| = |G_T| = |p|$, where p is a prime number. Let $e : G \times G \rightarrow G_T$ be a bilinear pairing on (G, G_T). The bilinear Diffie-Hellman problem (BDHP) is the following: Given P, aP, bP, cP, compute $e(P, P)^{abc}$.

The hardness of the BDHP implies the hardness of DHP in both G and G_T. First, if the DHP in G can be efficiently solved, then one could solve an instance of the BDHP by computing abP and then $e(abP, cP) = e(P, P)^{abc}$. Also, if the DHP in G_T can be efficiently solved, then the BDHP instance could be solved by computing $j = e(P, P)$, $j^{ab} = e(aP, bP)$, $j^c = e(P, cP)$ and then j^{abc}. Nothing else is known about the intractability of the BDHP, and the problem is generally assumed to be just as hard as the DHP in G and G_T [11, 12, 13].

The Modified Weil and Tate Pairings. Let G be a subgroup, with prime order p, of the curve's group of points $E(F_q)$. The modified Weil and Tate pairings yield a bilinear map $e : G \times G \rightarrow G_T$. The target group G_T is a subgroup of F_{q^k}, where k is a security multiplier that depends on the curve and on the group G [11, 12, 22]. DLP in G can be efficiently reduced to the DLP in G_T.

Theorem 1. *If there exists a bilinear map $e : G \times G \rightarrow G_T$, then the discrete log problem in G is no harder than the discrete log problem in G_T.*

Proof. Given $P \in G$ and $Q = xP \in G$, we can compute $i = e(P, P) \in G_T$, $j = e(P, Q) = e(P, xP) = e(P, P)^x \in G_T$. Thus, $\log_P Q = \log_i j$. So, it is clear that we can reduce the Discrete Log Problem in G to the Discrete Log Problem in G_T by using a discrete log solver for G_T to obtain x. This is called the MOV reduction [25]. □

4 Proposed Scheme

The proposed hidden digital signature scheme consists of three parties, the Requester (R), the Signer (S), and the Verifier (V). It is based on two protocols, the signing protocol and the verification protocol. The signing protocol runs two algorithms, the first one being the blinding algorithm, where is executed by R (the author of the message) and the other is the signing algorithm, executed in three steps by S. We assume that there is a trusted authority who establishes and manages the setup of the public key cryptosystem.

To implement our scheme we first need a security parameter that defines the level of bit strength that the

signature will provide. We then need to define groups G and G_T and a pairing $e : G \times G \rightarrow G_T$. To do this we pick an elliptic curve $E(F_q)$ with embedding degree k, where q is a prime power and it is the order of the finite field F_q. Also, p is a prime such that $p|\#E(F_q)$ where $\#E(F_q)$ is the order of the group $E(F_q)$, which is the number of points on an elliptic curve E over a field F, including the point at infinity. Moreover, the discrete log problem in Z_p^* is intractable.

We then randomly pick a point $P \in E(F_q)$, P is a point of order p in $E(F_q)$ and is called p-torsion point of the curve E. Let P be the generator of the group G and $e(P, P)$ the generator of the group G_T, which are cyclic groups of order p. G is a cyclic subgroup of $E(F_q)$ and G_T is a cyclic subgroup of $F_{q^k}^*$. We need cryptographic hash functions $H : \{0,1\}^* \rightarrow G$ and $h : \{0,1\}^* \rightarrow Z_p^*$, the security analysis will treat H and h as random oracles. Let M be the message and $m = h(M)$. The signer's secret key is $x \in Z_p^*$ and its public key is $Q = xP \in G$.

4.1 Signing Protocol

The signing protocol runs two algorithms, the blinding algorithm by the requester R and the signing algorithm by S.

4.1.1 Blinding Algorithm

The requester's aim is to get the signer's signature without disclosing the message content. At the same time the requester wants to make sure that the signer is the designated recipient of the blinded message. We can achieve this through double blinding the message by putting two locks on it. The first lock serves to blind the message from the signer. The second one is designated to the signer; he is the only one who can unlock it. These two locks can be done in one single mathematical operation; the requester will calculate $r = mQ$ and send it to the signer. Actually, the signer is the only one who can calculate $m \cdot P$ by using the multiplicative inverse of his secret key (the first lock), $r = m \cdot Q = m \cdot x \cdot P$. The resultant of this operation is still blind with respect to the signer's view and he needs to solve a hard problem, the discrete logarithm problem, to get m out of $m \cdot P$.

4.1.2 Signing Algorithm

Step 1. The signer receives$r = m \cdot Q$, and then he calculates $r^{'}$ as follows:

$$r^{'} = \frac{1}{x} \cdot r = \frac{1}{x} \cdot m \cdot Q = m \cdot P.$$

Step 2. The signer generates the signature parameter $z =< nounce||date||place >$ and then calculates $H(z)$.

Step 3. The signer generates the signature $(s,\ z)$ such that$s = \left(H(z) + r^{'}\right) \cdot \frac{1}{x} = \frac{H(z)+r^{'}}{x}$.

4.2 Verification Protocol

After revealing the message M, the signature is publicly verified by any verifier V using the bilinear pairing as follows:

$$e(s, Q) \stackrel{?}{=} e\left(H(z) + h(M) \cdot P,\ P\right).$$

The correctness of the proposed scheme is proven by:

$$\begin{aligned} L.H.S &= e(s, Q) \\ &= e(\frac{H(z) + r^{'}}{x}, x \cdot P) \\ &= e(\frac{H(z) + m \cdot P}{x}, x \cdot P) \\ &= e(H(z) + m \cdot P, P)^{\frac{1}{x} \cdot x} \\ &= e(H(z) + h(M) \cdot P, P) \\ &= R.H.S. \end{aligned}$$

5 Security Analysis

In this section, we review the known attacks on a digital signature scheme and the meaning of "breaking a signature scheme." Also, we discuss the notation of security of a blind digital signature scheme and a hidden digital signature scheme. Moreover, we explain the underlying cryptographic hard problem in the proposed scheme. This explanation is in Subsection 5.2 and it relates the proposed scheme to the Inverse computational Diffie-Hellman problem. In Subsection 5.3, the security of the proposed scheme is illustrated. This security analysis is twofold: First, it analyzes the blindness aspect in the proposed scheme. It then analyzes the non-forgeability aspect. We consider these two aspects, because the former is the core goal in the proposed scheme to hide the message and the latter is a mandatory property in any digital signature scheme. Also, in Subsection 5.3 we show the resistance of the proposed scheme against digital signature notorious attacks.

5.1 Security of Blind Signatures and Hidden Signatures

First, we illustrate some definitions from [14] that comprehend the kinds of attacks and the meaning of "breaking a signature scheme". Then, we discuss the notation of security of a blind digital signature scheme and a hidden digital signature scheme.

Attacks on a Digital Signature Scheme. Attacks are Key-Only Attacks and Message Attacks. In *Key-Only Attacks*, the adversary knows only the real signer's public key. In *Message Attacks*, the adversary is able to inspect some signatures corresponding to either a known or chosen-message before his attempt to break the scheme. Goldwasser, Micali, and Rivest [14] identified four

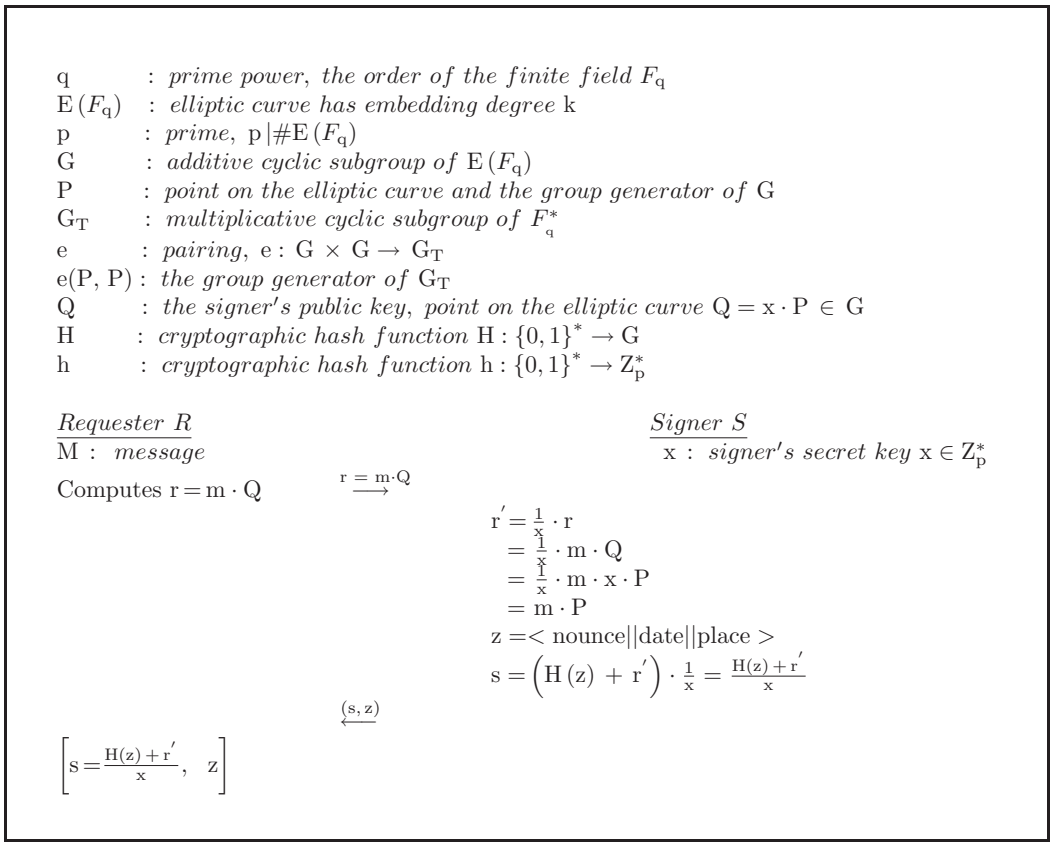

Figure 1: Signing protocol

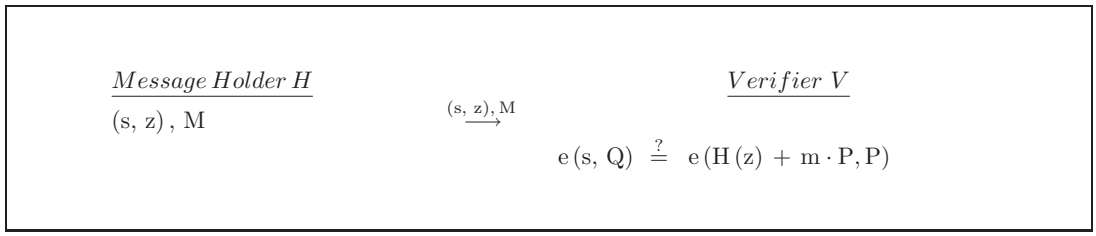

Figure 2: Verification protocol

kinds of message attacks grouped according to how the messages are chosen, and whose signatures the adversary sees. The following message attacks are listed in order of increasing severity, with the adaptive chosen-message attack being the most severe natural attack an adversary can mount: Known Message Attacks, Generic Chosen Message Attack, Directed Chosen Message Attack, and Adaptive Chosen Message Attack.

Breaking a Signature Scheme. The adversary breaks the signer S's signature scheme, if his attack allows him to do any of the following with a non-negligible probability (the kinds of "breaks" are listed in order of decreasing severity):

A Total Break: Compute S's secret trap-door information.

Universal Forgery: Find an efficient signing algorithm functionally equivalent to S's signing algorithm (based on possibly different but equivalent trap-door information).

Selective Forgery: Forge a signature for a particular message chosen a priori by the adversary.

Existential Forgery: Forge a signature for at least one message. The adversary has no control over the message whose signature he obtains, so it may be random or nonsensical. Rompel showed that signatures secured against existential adaptive chosen-message attacks can be based on general one-way functions [32]. This is illustrated in Theorems 2 and 3.

Theorem 2. *Under the assumption that one-way functions exist, one-way hash functions also exist.*

Theorem 3. *Under the assumption that one-way functions exist, there also exists a signature scheme which is secure against existential forgery under adaptive chosen message attacks.*

Proof. Refer to [32]. □

Security Notation of Blind and Hidden Signature Schemes. In [23], Jules *et al.* show how the security and blindness properties for blind digital signatures can be simultaneously defined and satisfied, assuming an arbitrary one-way trapdoor permutation family. They formally defined the notation of security of a blind digital signature scheme. Briefly, a blind digital signature scheme is secure if it satisfies both blindness and a non-forgeability property, non-forgeability meaning that after getting l signatures, it is infeasible for the adversary to compute l+1 signatures. Other formal definitions for blind digital signatures appear in [28, 29, 30] where the non-forgeability is called "one more" forgery. More definitions as (l, l+1)-forgery and the strong "one more"-forgery are introduced in [30]. In the context of blind signatures, the authors of [28] see that the definitions of security against the attacks, that we mentioned above, are no longer significant. In fact, they see that the existential forgery under an adaptively chosen message is somehow the basis for blind signatures. This is true in the case of strong blind signatures due to the fact that the requester modifies the signature in the unblinding phase and gets a different signed "message" than the one he received from the signer. Fortunately, this not the case in the proposed hidden signature scheme. The requester cannot modify the output of the signing protocol. Hence, we are still able to analyze the security of our scheme through the definitions in this section to achieve the non-forgeability property.

5.2 Inverse Computational Diffie-Hellman Assumption Security

In the signing algorithm, Subsection 4.1, the signature is constructed through calculating a point on the elliptic curve $E(F_q)$. In this arithmetic operation, the "s" part of the signature is the resultant of multiplying the term $(H(z) + r^{'})$ by $\frac{1}{x}$. Note that $(H(z) + r^{'})$ is a point on the elliptic curve $E(F_q)$ and the term $\frac{(H(z)+r^{'})}{x}$ is also a point on the elliptic curve $E(F_q)$. The only one who can perform this operation is the signer, who is the owner of the secret x. The construction of the signature is based on the Inverse Computational Diffie-Hellman assumption (InvCDH), which first appeared in [25]. To find the hardness equivalence of this signature with respect to Computational Diffie-Hellman assumption (CDH), first we review CDH. As in Section 3, Computational Diffie-Hellman is referred to as: on input P, aP, Q, outputs aQ.

An algorithm that solves the computational Diffie-Hellman problem is a probabilistic polynomial Time Turing Machine, on input P, aP, Q, outputs aQ with non-negligible probability. The Computational Diffie-Hellman assumption means that there is no such a probabilistic polynomial time Turing machine. This assumption is believed to be true for many cyclic groups, such as the prime subgroup of the multiplicative group of finite fields [10].

In [1], Bao *et al.* study various computational and decisional Diffie-Hellman problems by providing reductions between them in a high granularity setting. They considered the variations of Diffie-Hellman problems defined over some cyclic group with an explicit group structures. One variant of these computational Diffie-Hellman problem is the Inverse Computational Diffie-Hellman assumption. They showed that all variations of computational Diffie-Hellman problems are equivalent to the classic computational Diffie-Hellman problem if the order of an underlying cyclic group is a large prime. Also, they showed the same for variations of the decisional Diffie-Hellman problem except for the square decisional Diffie-Hellman problem, which did not prove or disprove the equivalence and they left it as an open problem.

The Inverse Computational Diffie-Hellman problem (InvCDH) is referred to as: P, xP, outputs $\frac{1}{x}P$. The value $\frac{1}{x}$ is the multiplicative inverse of $x \in Z_p^*$. An algorithm that solves the inverse computational Diffie-Hellman problem is a probabilistic polynomial Time Turing Machine, on input P, xP, outputs $\frac{1}{x}P$ with non-negligible probability. The Inverse computational Diffie-Hellman assumption means that there is no such a probabilistic polynomial Time Turing Machine.

Theorem 4. *[1] All variations of the computational Diffie-Hellman problem are equivalent to the classic computational Diffie-Hellman problem if the order of an underlying cyclic group is a large prime.*

Proof. Refer to [1] Section 2. □

To illustrate the importance of Theorem 4, let us analyze the following two attack scenarios and see how they fail:

Attack 1. In this attack an adversary requests the signer to sign a message $m = 1$. The signature will then be $s = \frac{(H(z)+P)}{x}$. He has to figure out $\frac{P}{x}$ given P, xP, because he aims to get $\frac{H(z)}{x}$ and according to Theorem 4, this is hard.

Attack 2. In this attack an adversary sends $r = P$ as a blinded message to the signer so the resultant signature will be $s = \frac{\left(H(z)+\frac{P}{x}\right)}{x}$. This signature cannot pass through the verification protocol because the adversary cannot provide either $\frac{P}{x}$ or a message that is equivalent to the secret $\frac{1}{x}$. From Theorem 4, this attack cannot succeed.

5.3 Proposed Scheme Security Aspects

In this subsection we discuss the security aspects of our proposed scheme. Our security analysis is twofold: first, it analyzes the blindness aspect in the proposed scheme. It then analyzes non-forgeability aspect. We consider these two aspects, because the former is the core goal in the proposed scheme to hide the message and the latter is a mandatory property in any digital signature scheme.

5.3.1 Blindness

"Blindness" means that the signer cannot know the content of the signed message as long as m is unrevealed by the message's owner, the requester of the signature. In the proposed scheme, the blindness algorithm depends on the hardness of the discrete logarithm problem in a group defined over an elliptic curve and there is no known algorithm that enables the efficient computation of discrete logs in this setup. Hence, from the signer's point of view, calculating m from mP is a hard problem and it is equivalent to solving a discrete logarithm problem in a group defined over an elliptic curve.

On the other hand, the adversary cannot even guess the mP, because he sees only mQ and to calculate m he needs to perform a total break of the cryptosystem by figuring out $\frac{1}{x}$ to get mP. After this, the adversary has to calculate m from mP which is a hard problem and is equivalent to solving a discrete logarithm problem in a group defined over an elliptic curve. As a result of this, m remains secret even if the cryptosystem is totally broken by compromising the secret key of the signer and the old signatures are still valid as if the message was undisclosed. In case of a total break of the cryptosystem, the signature can be verified by comparing the signature parameter z with the signer's database.

5.3.2 Non-forgeability

Theorem 5. *If $s^{'}$ is a random signature that has never been signed by the signer, then the verifier will accept $s^{'}$, as a valid signature for $H(z) + r^{'}$ with probability $1/p$.*

Proof. Since any group of prime order is cyclic, it follows that subgroup G is isomorphic to Z_p. □

The proposed scheme satisfies Theorem 4 and Theorem 5. Theorem 4 implies that the attacker cannot guess the signer's secret key from Q except with negligible probability equal to $1/p$. Also, Theorem 5 implies that the attacker cannot guess a random signature, $s^{'}$, on $H(z) + r^{'}$ except with negligible probability equal to $1/p$. Hence, both Theorems 4 and 5 prevent key-only attacks from succeeding. Moreover, Theorems 4 and 5 prevent a total break of the proposed signature scheme.

Theorem 6. *Given:*

1) *A message m_1.*
2) *Its corresponded signature (s_1, z_1) that is signed by the signer S, where z_1 is the signer's stamp on s_1.*
3) *A chosen stamp z_2, where $z_2 \neq z_1$, by the adversary.*

Then:
To find any random message m_2, where $m_2 \neq m_1$, that satisfies $s_1 = \frac{(H(z_2)+m_2P)}{x}$ is of a hardness equivalence to that of DLP in a group defined over an elliptic curve.

Given a signature (s_{1,z_1}) such that $s_1 = \frac{(H(z_1)+m_1P)}{x}$, the forgeability of s_1 is of a %hardness equivalence to that of DLP in a group defined over an elliptic curve to find any random m_2 that satisfies $s_1 = \frac{(H(z_2)+\%m_2P)}{x}$ for a chosen z_2.

Proof. Assume that $s_1 = \frac{(H(z_1)+m_1P)}{x} = \frac{(H(z_2)+m_2P)}{x}$, hence$(H(z_1) + m_1P) = (H(z_2) + m_2P)$ and$m_2P = H(z_1) - H(z_2) + m_1P$. □

Theorem 7. *Given:*

1) *A message m_1.*
2) *Its corresponding signature (s_1, z_1) that is signed by the signer S, where z_1 is the signer's stamp on s_1.*
3) *A chosen message m_2, where $m_2 \neq m_1$, by the adversary.*

Then:
To find any random stamp z_2, where $z_2 \neq z_1$, that satisfies $s_1 = \frac{(H(z_2)+m_2P)}{x}$ is of a hardness equivalence to that of breaking a secure hash function algorithm.

Given a signature (s_1, z_1) such that $s_1 = \frac{(H(z_1)+m_1P)}{x}$, the forgeability of s_1 is of a hardness equivalence to that of breaking a secure hash function algorithm to find any random z_2 that satisfies $s_1 = \frac{(H(z_2)+m_2P)}{x}$ for a chosen m_2.

Proof. Assume that $s_1 = \frac{(H(z_1)+m_1P)}{x} = \frac{(H(z_2)+m_2P)}{x}$, hence $(H(z_1) + m_1P) = (H(z_2) + m_2P)$ and $H(z_2) = m_1P - m_2P + H(z_1) = (m_1 + p - m_2)P + H(z_1)$. □

According to Theorems 2, 3, and 7, the proposed scheme is secure against an existential adaptive chosen message attacks. The adaptive chosen message attack is the most common type of attack in our application. To illustrate the role of Theorem 7, let us analyze the two following attack scenarios and see how they fail.

Attack 1. In this attack the requester aims to have a signature on m_2 using a legitimate signature s_1 that he has on m_1. Given that $s_1 = \frac{(H(z_1)+m_1P)}{x}$, the requester calculates $s_2 = \frac{m_2}{m_1}s_1 = \frac{m_2}{m_1} \cdot \frac{H(z_1)+m_1P}{x} = \frac{\frac{m_2}{m_1}H(z_1)+m_2P}{x}$, so the attack is reduced to find z_2 such that $H(z_2) = \frac{m_2}{m_1}H(z_1)$. From Theorem 7, this attack cannot succeed.

Attack 2. In this attack the requester aims to use two legitimate signatures, for which he knows the corresponding messages, to get a new signature on the sum of these two "hash" messages (the new message is likely to be meaningless). Given that $s_1 = \frac{(H(z_1)+m_1P)}{x}$ and $s_2 = \frac{(H(z_2)+m_2P)}{x}$, the requester calculates $s_n = s_1 + s_2 = \frac{(H(z_1)+m_1P)}{x} + \frac{(H(z_2)+m_2P)}{x}$, so the attack is reduced to find z_n such that $H(z_n) = H(z_1)+H(z_1)$. From Theorem 7, this attack cannot succeed.

Theorem 8. *Given a signature* $(s_1,\ z_1)$ *such that* $s_1 = \frac{(H(z_1)+m_1P)}{x}$, *the hardness of finding certain* m_2 *and* z_2 *pair such that* $s_1 = \frac{(H(z_2)+m_2P)}{x}$ *is hard enough as Theorem 6 or Theorem 7.*

Proof. Each of Theorems 6 and 7 has hardness degree in finding random m_2(or z_2) given z_2 (or m_2) respectively, adding more constraints on the randomness value of $m_2(z_2)$ but to be chosen in a certain way, does not reduce the problem hardness. Hence Theorems 6, 7, and 8 hold in the proposed scheme, the proposed scheme is secure against the selective forgery attack. □

6 Conclusion

If the signer has no interest in the message but the owner (the requester of the signature) has a special interest in the anonymity of the message, such as in notary services, we refer to such an application as a service with no conflict of interest. This is because the contributing parties in these applications are not interested in others' messages. Hence, a hidden signature can replace the strong blind signature in such an application. Also, the requester's ability to modify the signature might be an undesired property in some applications. For example, when a notary signs a client's (the requester) last will without knowing the content of it during the requester's lifetime. Later, when the lawyer reveals the message, the devisee or any entity can verify that the testament was signed by the notary. It will still be possible to check that the notary signed the testament even if the signature scheme has been broken in the meantime. The notary just looks in his list of signature parameters and compares the given signature parameters with his stored ones. This kind of application is also possible with weak blind signatures but not with strong signature schemes, because in the strong blind schemes the notary cannot find any relation between the given and the stored parameters. Other applications of hidden and weak blind signature schemes might be for pseudonymous credentials or anonymous access control.

Moreover, the proposed scheme allows the signer to insert a piece of temporal or spatial information in the signature that prevents the requester from modifying both these pieces of information as well as the signature. All of this is done without misusing the blindness property. Moreover, utilizing bilinear pairing in the proposed protocol makes use of the benefits of elliptic curve systems in terms of security and efficiency issues. The proposed scheme is proven to be secure against the existential adaptive chosen message attack. The existential adaptive chosen message attack is the natural and most severe attack in testament applications.

References

[1] F. Bao, R. Deng, and H. Zhu, "Variations of Diffie-Hellman problem", *Proceedings of Information and Communications Security*, LNCS 2836, pp. 301-312, 2003.

[2] M. Bellare and P. Rogaway, "The exact security of digital signatures - How to sign with RSA and Rabin," *Proceedings of Eurocrypt'96*, LNCS 1070, pp. 399-416, 1996.

[3] A. Boldyreva, "Efficient threshold signature, multisignature and blind signature schemes based on the Gap-Diffie-Hellman-group signature scheme," *Proceedings of Practice and Theory in Public Key Cryptography - PKC'2003, LNCS 2567*, pp. 31-46, 2003.

[4] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *Proceedings of CRYPTO 2001*, LNCS 2139, pp. 213-229, 2001.

[5] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *Advances in Cryptology - ASIACRYPT 2001*, LNCS 2248, pp. 514-532, 2001.

[6] D. Boneh, "The decisional diffie-hellman problem," *Third Algorithmic Number Theory Symposium*, pp. 48-63. 1998.

[7] J. Camenisch, J. Piveteau, M. Stadler, "Blind signatures based on discrete logarithm problem," *Advances in cryptology, EUROCRYPT'94*, LNCS 950, pp. 428-432, 1994.

[8] D. Chaum, "Blind signatures for untraceable payments," *Advances in Cryptology - Crypto '82, Springer-Verlag*, pp. 199-203, 1983.

[9] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 10, pp. 1030-1044, 1985.

[10] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 6, no. 2, pp. 644-654, Nov. 1976.

[11] G. Frey, M. Muller, and H. Ruck, "The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 171-179, 1999.

[12] S. Galbraith, K. Harrison, and D. Soldera, "Implementing the Tate pairing," *Proceedings of ANTS V*, LNCS 2369, pp. 324-37, 2002.

[13] P. Gaudry, F. Hess, and N. Smart, "Constructive and destructive facets of weil descent on elliptic curves," Journal of Cryptology, vol. 15, no. 1, pp. 19-46, 2002.

[14] S. Goldwasser, S. Micali, and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM Journal of Computin*, vol. 17, no. 2, pp. 281-308, 1988.

[15] P. Horster and H. Petersen, *Classification of blind signature schemes and examples of hidden and weak blind signatures,* Technical Report TR-94-1, May 1994, Theoretical Computer Science and Information Security, Department of Computer Science, University of Technology Chemnitz-Zwickau, Germany, Presented at the Rump Session of Eurocrypt '94, 6 pages, Perugia, Italy, 1994.

[16] P. Horster, M. Michels, and H. Petersen, *Hidden signature schemes based on the discrete logarithm problem and related concepts,* Technical Report TR-94-40-R, Theoretical Computer Science and Information Security, Department of Computer Science, University of Technology Chemnitz-Zwickau, Germany, Apr. 1995.

[17] P. Horster, M. Michels, H. Petersen, and H. Petersen, Meta-Message recovery and Meta-Blind signature schemes based on the discrete logarithm problem and their applications, Advances in Cryptology - ASIACRYPT'94, LCNS 917, pp. 224-237, Berlin, 1995.

[18] L. Han, "Cryptanalysis of the blind signatures based on the discrete logarithm problem," *IEE Electronic Letters*, vol. 31, no. 14, pp. 1136-1137, 1995.

[19] P. Horster and H. J. Knobloch, "Discrete logarithm based protocols," *Advances in Cryptology: Proceedings Eurocrypt'91*, LNCS 547, pp. 399-408, 1992.

[20] P. Horster, M. Michels, and H. Petersen, "Meta-ElGamal signature schemes," *Proceedings 2nd ACM conference on Computer and Communications Security*, pp. 96-107, Nov. 2-4, 1994.

[21] A. Joux, "A one-round protocol for tripartite Diffie-Hellman", *Algorithm Number Theory Symposium - ANTS-IV*, LNCS 1838, pp. 385-394, 2000.

[22] A. Joux and K. Nguyen, "Separating decision Diffie-Hellman from Diffie-Hellman in cryptographic groups," *Springer Journal of Cryptology*, vol. 16, pp. 239-247, 2003.

[23] A. Juels, M. Luby, and R. Ostrovsky, "Security of blind digital signatures (Extended Abstract)," *Advances in Cryptology-Crypto 1997*, LNCS 1294, pp. 150-164, 1997.

[24] L. Martin, *Introduction to Identity-Based Encryption*, chapter 3, Information Security and Privacy Series, Artech House, INC. 2008.

[25] A. Menezes, T. Okamoto, and P. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field," *IEEE Transactions on Information Theory*, vol. 39, no. 5, pp. 1639-46, 1993.

[26] T. Okamoto and D. Pointcheval, "The gap problems: A new class of problems for the security of cryptographic primitives," *Proceedings of PKC 2001*, LNCS 1992, pp. 104-18, 2001.

[27] B. Pfitzmann and A. Sadeghi, "Anonymous fingerprint with direct non-repudiation," *Advances in Cryptology - ASIACRYPT' 2000*, LNCS 1976, pp. 401-414, 2000.

[28] D. Pointcheval and J. Stern, "Provably secure blind signature schemes," *Advances in Cryptology - Asiacrypt 1996*, LNCS 1163, pp. 252-265, 1996.

[29] D. Pointcheval and J.Stern, "Security proofs for signature schemes," *Advances in Cryptology - Eurocrypt 1996*, LNCS 1070, pp. 387-398, 1996.

[30] D. Pointcheval and J.Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology,* vol. 13, no. 3, pp. 361-396, 2000.

[31] D. Pointcheval and J.Stern, "Provably secure blind signature schemes," *Advances in Cryptology - Asiacrypt 1992*, LNCS 1163, pp. 252-265, 1996.

[32] J. Rompel, "One-way functions are neessary and sufficient for secure signatures," *STOC 90: 22nd Annual ACM Symposium on Theory of Computing*, pp. 387-394, 1990.

[33] Z. Zhao, "D-based weak blind signature from bilinear pairings," *International Journal of Network Security*, vol. 7, no. 2, pp. 265-268, 2008.

Mohamed Rasslan received the B.Sc. and M.Sc. degrees from Cairo University and Ain Shams University, Cairo, Egypt, in 1999 and 2006 respectively. During 1999-2006, he worked at Egyptian Cabinet Information and Decision Support Center, TE Data S.A.E, and the Egyptian Ministry of Communications and Information Technology. He is currently a Research Assistant at the Electrical and Computer Engineering Department, Concordia University, Montreal, Canada. His main research interests are in the area of analysis and design of secure and private electronic communications protocols.