J. Appl. Math. & Informatics Vol. **29***(2011), No. 3 - 4, pp. 869 - 877*
Website: http://www.kcam.biz

ON EFFICIENT TWO-FLOW ZERO-KNOWLEDGE IDENTIFICATION AND SIGNATURE

YOUNG WHAN LEE

Abstract. In this paper, we propose an efficient two-flow zero-knowledge blind identification protocol on the elliptic curve cryptographic (ECC) system. A. Saxena et al. first proposed a two-flow blind identification protocol in 2005. But it has a weakness of the active-intruder attack and uses the pairing operation that causes slow implementation in smart cards. But our protocol is secure under such attacks because of using the hash function. In particular, it is fast because we don't use the pairing operation and consists of only two message flows. It does not rely on any underlying signature or encryption scheme. Our protocol is secure assuming the hardness of the Discrete-Logarithm Problem in bilinear groups.

AMS Mathematics Subject Classification : 94A60, 94A62, 68U20.
Key words and phrases : Identification, Signature, Zero-Knowledge, Bilinear pairing.

1. Introduction

An identification protocol is an interactive protocol between the prover, *Alice* or A and verifier, *Bob* or B in which the prover tries to identify itself to the verifier by demonstrating knowledge of a certain key associated with the prover. In the symmetric (secret key) setting, the key is shared between prover and the verifier, whereas in the asymmetric (public key) setting, the key is the private key of the prover. In this paper we are interested in the public key setting. There are many identification protocols using zero-knowledge proofs [2, 6, 7, 8, 9, 10]. A. Saxena et al. [11] introduce the notion of bounded-prover zero-knowledge proofs which require only two rounds and can be considered perfectly zero-knowledge under certain interactivity assumptions. Their protocol uses bilinear pairings and can be encapsulated in smart cards disguised for elliptic curve cryptography (ECC). But, unfortunately pairing implementation attempts in limited devices such as agents reveal that code may be slow, resource consuming and tricky to

Received August 11, 2010. Revised March 14, 2011. Accepted April 5, 2011.

program, although pairing is a cubic-time implementation. Also their scheme has a weakness of the active-intruder attack. To improve these weaknesses, we propose a new zero-knowledge blind identification protocol. We do not use bilinear pairings in identification and signature. And we prove the protocol is secure assuming the hardness of the Discrete-Logarithm Problem in bilinear groups. The organization of the paper is as follows. In Section 2, we present hard problems such as DLP and DHP, and the preliminaries of bilinear pairings. In section 3, we investigate backgrounds, and give an example of the active-intruder attack on Saxena et al.'s scheme. In section 4, we present our contributions. In Section 5, we propose our new two-round zero-knowledge blind identification and then in Section 6 we prove the security of the proposed protocol. In Section 7, we deal with other extensions such as hidden signatures. Finally, a conclusion is given in Section 8.

2. Hard Problems

Let G_1 be an additive cyclic group of the prime order q and G_2 be the multiplicative cyclic group of the same order. Practically we think of G_1 as a group of points on an elliptical curve on Z_q^*, and G_2 as a subgroup of the multiplicative group of a finite field $Z_{q^k}^*$ for some $k \in Z_q^*$. Let P be a generator of G_1 .

We define the following problems in G_1.

(1) Discrete-Logarithm Problem (DLP) : Given $P, Q \in G_1$, find an integer $a \in Z_q^*$ such that $aP = Q$.
(2) Diffie-Hellman Problem (DHP) : Given $P, xP, rxP \in G_1$ for unknowns $x, r \in Z_q^*$, compute $rP \in G_1$.

The cryptology using pairings is based on the existence of efficiently computable non-degenerate bilinear maps (or pairings) which can be abstractly described as follows.

A map $\hat{e} : G_1 \times G_1 \to G_2$ is called *bilinear pairing* if $\hat{e}$ satisfies the following properties:

(1) Bilinearity : For all $P, Q \in G_1$ and $a, b \in Z_q^*$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$
(2) No-degeneracy : $P \neq 0 \Rightarrow \hat{e}(P, P) \neq 1$
(3) Computability : There is an efficient algorithm to compute $\hat{e}(P, Q)$ for all $P, Q \in G_1$

Note that modified Weil pairing and Tate pairing are examples of bilinear pairings [3]. Without going into the details of generating suitable curves, we may assume that $q \approx 2^{171}$ so that the fastest algorithms for computing discrete logarithms in G_1 take about 2^{85} iterations [11].

3. Background

In this section, we introduce a two-round identification scheme using a public key cryptosystem, which proposed by A. Saxena, B. Soh and S. Priymak [12], D. R. Stinson and J. Wu [15] and the author [10]. Assume that *Alice* and *Bob* are

two users and *Alice* wants to identify herself to *Bob*. We only consider one-way identification and ignore the case of *Bob* identifying himself to *Alice*. A round of a protocol involves the exchange of one message. A sequence of two synchronous message transmissions constitutes two separate rounds, while any number of asynchronous messages is the part of the same round. A single message passing is a one-round protocol.

(1) The SSP (A. Saxena, B. Soh and S. Priymak [11]) Scheme:
1. B chooses $r \in Z_q$ uniformly at random and compute $R = rY$ and $U = r^2P$. Then B sends $< R, U >$ to A.
2. After receiving $< R, U >$, A computes $\frac{1}{x}R$. A rejects and stops if $\hat{e}(\frac{1}{x}R, \frac{1}{x}R) \neq \hat{e}(U, P)$; otherwise A generates $Q \in G_1$ and computes $Z = V + xQ$. And then A sends $< Z, Q >$ to B.
3. After receiving $< Z, Q >$, B verifies $< Z, Q >$; If $\hat{e}(Z - rP, P) = \hat{e}(Q, Y)$, then B accepts : otherwise, B rejects.

(2) The SW (D.R. Stinson, J.Wu [12]) Scheme:
1. B chooses $r \in Z_q$ uniformly at random and compute rP and $h(rxP)$. where $h : G_1 \to \{0,1\}^k$ is a hash function. Then B sends $< rP, h(rxP) >$ to A.
2. After receiving $< rP, h(rxP) >$, A verifies if $rp \in G_1$ and compute xrP. If $rP \in G_1$ and $h(xrP) = h(rxP)$ then A sends xrP to B; otherwise A reject and stops.
3. After receiving xrP , B verifies xrP ; If $xrP = rxP$, then B accepts; otherwise, B rejects.

(3) The CL (B.M. Choi, Y.W. Lee [10]) scheme:
1. B chooses $r \in Z_q$ uniformly at random and compute

$$V = \hat{e}(rP, xP) = C^{rx}, \quad W = \hat{e}(rP, P)$$

and $h(V)$ where $h : G_1 \to \{0,1\}^k$ is a hash function. Then B sends $< W, h(V) >$ to A.
2. After receiving $< W, h(V) >$, A reject and stops if $h(V) \neq h(W^x)$, or $W \notin G_2$: otherwise A choose $z \in Z_q$ and compute $X = W^{x+zx}$ and $T = W^{zx}$. Then A send $< h(X), T >$ to B.
3. After receiving $< X, T >$, B verifies $< X, T >$; If $X = TV$, then B accepts : otherwise, B rejects.

(4) Active-intruder Attack on SSP Scheme:
Informally, an active adversary is the one who alters, injects, drops or diverts messages between the prover and the verifier. Note that there are three approaches to handling this definitional issue [1, 5, 12]. D. R. Stinson, J. Wu defined a successful active-intruder attack. In an active-intruder attack, the adversary is successful if the (honest) verifier accepts in a session after the adversary becomes active in the same session [12]. We give an example of the active-intruder attack on SSP scheme as follows:

We use simple figures and notations to illustrate the SSP protocol and corresponding active-intruder attacks on it. Let r be a random number chosen by B, X a random number chosen by A, and O any attacker. All computations take place in a relevant group.

SSP Scheme: x is secret key and xP is public key.

$$A \xleftarrow{<R=rxP, U=r^2P>} B$$

$$A \xrightarrow{<Z=\frac{1}{x}R+xQ, Q>} B$$

A verifies that $\hat{e}(\frac{1}{x}R, \frac{1}{x}R) = \hat{e}(U, P)$ and accepts. Also verifies that $\hat{e}(Z - rP, P) = \hat{e}(Q, xP)$ and accepts.

Attack : The active-intruder attack is possible.

$$A \xleftarrow{<2R=2rxP, 4U=4r^2P>} O \xleftarrow{<R=rxP, U=r^2P>} B$$

$$A \xrightarrow{<Z=\frac{1}{x}2R+xQ, Q>} O \xrightarrow{<\frac{1}{2}Z, \frac{1}{2}Q>} B$$

A verifies that

$$\hat{e}(\frac{1}{x}2R, \frac{1}{x}2R) = \hat{e}(2rP, 2rP) = \hat{e}(P, P)^{4r^2} = \hat{e}(4U, P)$$

and accepts. B verifies that

$$\hat{e}(\frac{1}{2}Z - rP, P) = \hat{e}(\frac{1}{2}xQ, P) = \hat{e}(Q, P)^{\frac{x}{2}} = \hat{e}(\frac{1}{2}Q, xP)$$

and accepts.

4. Our Contribution

In this paper, we propose a new 2-flow identification protocol for smart cards using a public key cryptosystem. Our proposed protocol has several advantages.

(1) For a computationally limited device such as a smart card, the prover and the verifier in our protocol do not use bilinear pairings on ECC.
(2) Our protocol is secure assuming only the hardness of the Discrete Logarithm Problem in bilinear groups. Note that the SSP scheme and the SW scheme need another assumption such as the hardness of the DHP, EDHP or LDHP [11, 12].
(3) The SSP scheme has a weakness of the active-intruder attack, but our scheme does not.
(4) The SW scheme is only a zero-knowledge identification protocol. But our scheme has other extensions such as blind identifications, signatures and on-line credit card payments.

5. Our New Two-flow Zero-knowledge Identification Scheme

5.1. Initial Setup. We assume the existence of a trusted authority, denoted by TA, who will issue certificates for all potential participants in the scheme. The initial setup for our scheme as follows:

Protocol 5.1: Zero-knowledge Scheme Setup

Input: Security parameter $k \in Z^+$.

(1) The TA generates a prime q, two groups G_1, G_2 of order q.
(2) The TA chooses a random generator $P \in G_1$.
(3) The TA publishes a hash function $h : G_1 \to \{0,1\}^k$.
(4) The TA publishes the system parameters $< q, G_1, G_2, P, h >$.
(5) Each potential prover A chooses a private key x uniformly from Z_q^* at random, computes xP and registers xP as A's public key.

5.2. Proposed Protocol Description. In a session of the scheme, the prover A tries to convince the verifier B of A's identity. B accepts only if A respond to B's challenge in an appropriate way. The steps in a session of our scheme as follows:

Protocol 5.2: A 2-flow Zero-knowledge scheme

(1) The verifier B chooses $r \in Z_q^*$ uniformly at random, and computes $V = rxP$, $W = rP$ and $h(V)$. Then B sends $< h(V), W >$ to the prover A.
(2) After receiving $< h(V), W >$, A rejects and stops if $h(V) \neq h(xW)$, or $W \notin G_2$; otherwise A chooses $z \in Z_q$, and compute $X = xW + zxW$, $h(X)$ and $T = zxW$. Then A sends $< h(X), T >$ to B.
(3) After receiving $< h(X), T >$, B accepts if $h(X) = h(T + V)$; otherwise B rejects.

$$A \xleftarrow{<W=rP, \quad h_1=h(rxP)>} B$$

$$A \xrightarrow{<T=zxW, \quad h_2=h(xW+zxW)>} B$$

5.3. Completeness. It is straightforward to prove that Protocol 5.2 is complete. Let A and B be both honest. After receiving the challenge $< h(V), W >$, A checks to see if $h(V) \equiv h(xW)$. Since $V = rxP = xW$, A accepts and sends the response $< h(X), T >$ to B. Then B checks to see if $h(X) = h(T + V)$. Since

$$T + V = zrxW + rxP = xW + zxW = X,$$

B also accepts.

6. Security of the Proposed Protocol

In this section, we prove that the above protocol is perfect zero-knowledge.

6.1. Dishonest Prover Zero-knowledge. Assuming an honest verifier, we must show that a dishonest prover cannot succeed except with a negligible probability. Given $xP, h(V)$ and W , the task of a dishonest prover is to compute a pair $< h(X), T >$ such that $X = xW + zxW$. We show that this is an instance of the DLP as following theorem.

Theorem 6.1. (Soundness 1) *Assume that the DLP is hard. Then it is hard for the dishonest prover to construct a pair* $< h(X), T >$ *with* $X = T + V$.

Proof. The dishonest knows the system parameters $< q, G_1, G_2, P, h >$ and the transcript $< h(X), T >$. Also he knows the public keys xp and rP of the prover and verifier, respectively, but he does not know r and x in Z_q^*. Now assume that $X = sW + zsW$ and $T = zsW$ for $s, z \in Z_q^*$. If $X = T + V$, then

$$sW + zsW = srP + zsrP = zsrP + rxP.$$

Thus $srP = xrP$ and so $sW = V$. That is, this states that to compute a pair $< h(X), T >$ with $X = T + V$ for unknown for unknowns $r, x \in Z_q^*$ is to choose s satisfying $sW = V$ for the known $V, W \in G_1$. This is the Discrete Logarithm Problem and thus it is hard for a dishonest prover to construct $< h(X), T >$ with $X = T + V$. □

6.2. Dishonest Verifier Zero-knowledge. A dishonest verifier will generate $< h(V), W >$ with $h(V) = h(xW)$ non-uniformly. In order words, a dishonest verifier will not know r corresponding to V. To prove Zero-knowledge in this case, it is enough to prove that the probability of a dishonest verifier succeeding is the probability solving the Discrete Logarithm Problem.

Theorem 6.2. (Soundness 2) *Assume that the DLP is hard and* $h(\cdot)$ *is random oracle. Then it is hard for a dishonest verifier to construct* V *such that* $h(V) = h(xW)$ *for given* W, P, xP.

Proof. To construct V, a dishonest verifier must chooses $r' \in Z_q^*$ such that $V = rxP = r'xP = xW$ for unknowns $r, x \in Z_q^*$. That is, he chooses $r' \in Z_q^*$ such that $x'R = Q$ for $R = xP$ and $Q = xW$. This is the Discrete-Logarithm Problem and so it is hard. □

The knowledge of W and $h(V)$ does not give a dishonest prover any additional advantage in solving this DLP instance because deciding if $h(V) = h(xW)$ is an instance of the DLP as Theorem 2. Thus, the proof is sound from a verifier's view as long as the DLP is intractable.

6.3. Honest Verifier Zero-knowledge. The transcript consists of the messages exchanged between the two parties. In Theorem 6.3, we construct a simulator that can generate an accepting transcript $\{h(V), W, h(X), T\}$ without interaction with a prover and then show that the simulated and real distributions are identical. Thus our protocol is perfect zero-knowledge for an honest verifier.

Theorem 6.3. *Protocol 5.2 is perfect zero-knowledge for an honest verifier.*

Proof. The set $\Im$ of real transcripts obtained by a prover and an honest verifier consists of all transcripts obtained as the the following form:

$$\begin{aligned}\Im &= \{h(V), W, h(X), T\} \\ &= \{h(xrP), rP, h((x+zx)rp, zxrP \big| z, x \in Z_q^*\}\end{aligned}$$

where r is chosen by the verifier uniformly at random from Z_q^* and also x, z is chosen by the prover uniformly at random from Z_q^*.

Now let $\hat{\Im}$ be the set of simulated transcripts can be constructed by the verifier as follows; The verifier chooses r and α uniformly at random from Z_q^* and computes the simulated transcript

$$\hat{\Im} = \{h(\alpha P), rP, h(\alpha rP + \alpha\beta P \big| \alpha, \beta \in Z_q^*\}.$$

Since the random numbers x, z and α, β in Z_q^* have identical probability distribution. $\Im$ and $\hat{\Im}$ have identical probability distributions. Therefore the protocol is perfect zero-knowledge for an honest verifier. □

6.4. Passive Adversary Blindness. An inherent property of our protocol is *passive adversary blindness* which informally implies that no polynomially bounded adversary has a non-negligible advantage in deciding the honesty of the participants in the protocol. Assuming that the DLP is intractable, it is impossible for a passive adversary to decide the honesty of the verifier: given $< P, xP, W, h(V) >$, deciding if $V = xW$ is an instance of the DLP. Similarly it is impossible for a passive adversary to decide the honesty of the prover: given $< P, xP, h(X), T >$, deciding if $X = T + V$ is an instance of the DLP.

6.5. Knowledge Extractor. Let $L_1 = \{< X, T > | X = T + V\}$. Then a prover ID_B essentially proves knowledge of the witness $< h(X), T >$ in L_1 using the shared string

$$< P, xP, rP, h(rxP) >$$

and unknown $x \in Z_q^*$. Assume that a dishonest prover ID^* is able to make any verifier accept. Then given $< P, xP, rP, h(rxP) >$, ID^* can always output a pair $< h(X'), T' >$ such that $X' = T' + V$ From simulating the honest verifier itself and the hardness of DLP, It is impossible to make $< h(X'), T' > \in L_1$ for unknown $x \in Z_q^*$. Thus our protocol is a *proof of knowledge*

7. Other Extension

In this section we deal with other extensions such as hidden signatures and credit payments.

7.1. Hidden Signatures. When user A identifies to the server B, A can also send plain text message along with hidden signature such that B can extract the signature.

Protocol 7.1 : Hidden Signature Scheme

(1) Initialization : The server B asks user A to identify itself by sending the challenge $< h(V), W >$ in the first step of Protocol 5.2.
(2) Signing : Let $M \in G_1$ be the message to be signed and $H(M) = \beta$, where $H : G_1 \to Z_q^*$ is a hash function. A computes xW and check that $h(V) = h(xW)$. And then A choose $z \in Z_q^*$ randomly and compute $h(X) = h(\beta T + \beta V) = h(zx\beta W + x\beta W)$ and $T = zxW$. The pair $<< X, T >, M >$ is sent to B.
(3) Verification : After receiving $<< X, T >, M >$, B extracts the signature $S = T + V$. The verification condition is $h(X) = h(\beta S)$.

7.2. Credit Payments. As A. Saxena et al. proposed the secure of a anonymous seller credit payments, our secure Zero-knowledge identification and hidden signature scheme can be applied to them without the active-intruder attack.

8. Conclusion

In this paper, we proposed a new zero-knowledge blind identification protocol for smart cards. Only based on the DLP assumption, it is secure in random oracle model. Also in our protocol the prover and verifier do not use bilinear pairings. Thus agents with our scheme need not have devices for bilinear pairings. Under the methods of security proof given by Stinson and Wu [12], our protocol is secure against active-intruder attacks but Saxena et al.' scheme [11] has a weakness of them. As we compare SSP and SW scheme with ours, we know that the performance of our scheme is more useful as following table.

Table1: Compare of Performances

Performance	SSP Scheme	SW Scheme	Our Scheme
No. of flows	2	2	2
No. of system parameters	7	5	5
Active intruder attack	Not secure	Secure	Secure
Extend to hidden signature	Extend	Not extend	Extend
Extend to seller payments	Extend	Not extend	Extend
Need of hard problems	DLP, DHP	DLP, DHP	DLP

References

1. M. Bellare and P. Rogaway, *Entity authentication and key distribution*, Lec-ture Notes in computer Science **773** (1994), 232-149.
2. M. Bellare and O. Goldreich, *On defining proofs of knowledge*, Lecture Notes in computer Science **740** (1993), 390-420.

3. D. Boneh, B. Lynn, and H. Shacham, *Short signatures from the Weil pairing*, In ASIACRYPT '01: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security, London, UK, Springer-Verlag, (2001), 514-532.
4. D. Boneh and M.K. Franklin, *Identity-based encryption from the Weil pairing*, In CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, Springer-Verlag, (2001), 213-229.
5. W. Diffie, P.C. van Oorschot and M.J. Wiener, *Authentication and Authenticated key exchanges* Designs, Codes and Cryptography **2** (1992), 107-125.
6. U. Feige, A. Fiat, and A. Shamir, *Zero knowledge proofs of identity*, J. Cryptology **1** (1988), 77-94.
7. A. Fiat and A. Shamir, *How to prove yourself: practical solutions to identification and signature problems*, Advances in Cryptology, Lecture Notes in Computer Science **263** (1987), 186-194.
8. O. Goldreich, S. Micali, and A. Wigderson, *Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems*, J. ACM **38** (3) (1991), 690-728.
9. Y. W. Lee and B. M. Choi, *Intelligent ID-based threshold systems by an encryption and decryption from bilinear pairing*, Lecture Notes in Artificial Intelligence, 9th International Conference, KES LNAI **3682** (2005) 1022-1028.
10. B. M. Choi and Y. W. Lee, *Secure identification and signature using Zero-knowledge proofs and bilinear pairings*, J. Chungcheong Math. Soc. **21** (2008) 403-411.
11. A. Saxena, B. Soh and S. Priymak, *Zero-Knowledge blind identification for smart cards using bilinear pairings*, Cryptology e-Print Archive, Report **343**, (2005).
12. D.R. Stinson and J. Wu, *An efficient and secure two-flow zero-knowledge identification protocol*, Cryptology e-Print Archive, Report **337**, (2006).

Young Whan Lee received M.Sc. from Chungnam National University and Ph.D at Chungnam National University. Since 1988 he has been at Daejeon University. His research interests include cryptology and information security.

Department of Computer Hacking and Information Security, College of Natural Sciences, Daejeon University, Daejeon, 300-716, Korea.
e-mail: ywlee@dju.ac.kr