JOURNAL OF THE
CHUNGCHEONG MATHEMATICAL SOCIETY
Volume **21**, No. 3, September 2008

SECURE IDENTIFICATION AND SIGNATURE USING ZERO-KNOWLEDGE PROOFS AND BILINEAR PAIRINGS

Byung Mun Choi* and Young Whan Lee**

Abstract. In 2005, A. Saxena, B. Soh and S. Priymak [10] proposed a two-flow blind identification protocol. But it has a weakness of the active-intruder attack and uses the pairing operation that causes slow implementation in smart cards. In 2008, Y. W. Lee [9] made a method of the active-intruder attack on their identification scheme and proposed a new zero-knowledge blind identification protocol for smart cards. In this paper, we give more simple and fast protocols than above protocols such that the prover using computationally limited devices such as smart cards has no need of computing the bilinear pairings. Computing the bilinear pairings is needed only for the verifier and is secure assuming the hardness of the Discrete-Logarithm Problem (DLP).

1. Introduction

A. Saxena, B. Soh and S. Priymak [10] proposed a two-flow blind identification protocol using 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 smart cards reveal that code may be slow, resource consuming and tricky to program, although pairing is a cubic-time implementation. Note that an identification protocol is an interactive protocol between the prover and verifier, in which the prover tries to identify itself to the verifier by demonstrating knowledge of a certain key associated with the prover. In the secret key setting, the key is shared

Received July 31, 2008; Revised August 21; Accepted August 22, 2008.
2000 Mathematics Subject Classification: Primary 39B72, 39B22.
Key words and phrases: identification, signature, zero-knowledge, smart card, bilinear pairing.

between prover and the verifier, whereas in the public key setting, the key is the private key of the prover. In this paper we are interested in the public key setting. In 2008, Y. W. Lee [9] made a method of the active-intruder attack on their identification scheme and propose a new zero-knowledge blind identification protocol for smart cards.

In this paper, we give simple and fast protocols such that the prover using computationally limited devices such as smart cards has no need of computing the bilinear pairings. Computing the bilinear pairings is needed only for the verifier and is secure assuming the hardness of the Discrete-Logarithm Problem. The organization of the paper is as follows; In Section 2, we present the preliminaries of bilinear paring and background. In Section 3, we propose our new secure identification protocol and then in Section 4, we prove the security of the proposed protocol. In Section 5, we propose a hidden signatures. Finally, a conclusion is given in Section 6.

2. Bilinear pairings and background

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; 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. 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, 4]. 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 [12]. 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 $rP \in G_1$.

In this section, we introduce a two-round identification schemes using a public key cryptosystem, which proposed by A. Saxena, B. Soh and S. Priymak [10] and Y. W. Lee [9]. 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 part of the same round. A single message passing is a one-round protocol.

1. The SSP (A. Saxena, B. Soh and S. Priymak [10]) 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 YWL (Y. W. Lee [9]) Scheme:
   (1) B chooses $r \in Z_q$ uniformly at random and compute $V = \hat{e}(rxP, xP) = C^{rx^2}, W = \hat{e}(rP, xP) = C^r x$ and $h(V)$. Then B sends $< h(V), W >$ to A.
   (2) After receiving $< h(V), W >$, A rejects and stops if $h(V) \neq h(W^x)$, or $W \notin G_2$; otherwise A chooses $z \in Z_q$ and computes $X = W\frac{1}{x}C^{x^3z}$ and $T = W^{x^2z}$. And then A sends $< X, T >$ to B.
   (3) After receiving $< X, T >$, B accepts if $X = C^{rT\frac{1}{r}}$: otherwise, B rejects.

Informally, an active adversary is the one who alters, injects, drops and/or diverts messages between the prover and the verifier. Note that there are three approaches to handling this definitional issue [1, 5, 11]. D. R. Stinson, J. Wu defined a successful active-intruder attack as follow: 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 [11]. Young Whan Lee [9] suggested an example of the active-intruder attack on SSP scheme.

In this paper, we propose a simple and fast 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 in our protocol does not use bilinear pairings and only the verifier uses them. and Our protocol is more simpler and faster than YWL scheme.
2. Our protocol is secure assuming only the hardness of the Discrete-Logarithm Problem in bilinear groups as well as YWL scheme. Note that the SSP scheme needs another assumption such as the hardness of the DHP, EDHP or LDHP [10].
3. The SSP scheme has a weakness of the active-intruder attack, but our scheme does not as well as YWL scheme.

3. Our new secure blind identification

3.1. Initial identification setup

Let TA be the trusted authority, by assuming the existence of a trusted authority, who will issue certificates for all potential participants in the protocol. The initial setup for our protocol as following;

Protocol 3.1: Initial identification scheme setup

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

1. The TA generates a prime q, two groups G_1, G_2 of order q and an admissible bilinear map $\hat{e} : G_1 \times G_1 \to G_2$.
2. The TA chooses a random generator $P \in G_1$, a random $s \in Z_q^*$ and sets $P_{pub} = sP$.
3. The TA publishes a hash function $h : G_2 \to \{0,1\}^k$.
4. The TA computes C such that $C = \hat{e}(P, P)$, and publishes the system parameters $< q, G_1, G_2, P, P_{pub}, \hat{e}, C, h >$.
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.

3.2. Protocol description

In a session of the protocol, 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 following;

Protocol 3.2: A 2-flow new identification protocol

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

3.3. Completeness

Suppose A and B are both honest.

1. After receiving the challenge $< h(V), W >$, A can check to see if $h(V) = h(W^x)$, because $W^x = C^{rx} = V = C^{rx}$. Thus A can accept or reject. If A accepts then A sends the response $< h(X), T >$ to B.
2. Then B can checks to see if $h(X) = h(V)$, because $X = TV$. Thus B also can accept or reject.

4. Security of our new identification protocol

In this section, we prove that our new identification protocol is perfect zero-knowledge.

4.1. Soundness

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 $W = \hat{e}(rP, P) = C^r, X = W^{x+zx}$, and $T = W^{zx}$. We show that this is an instance of the DLP in the following Theorem 4.1. The knowledge of W and $h(V)$ does not give a dishonest prover any additional advantage in solving this DLP instance because deciding if is an instance of the DLP as the Theorem 4.1. Thus, the proof is sound from a verifier's view as long as the DLP is intractable.

THEOREM 4.1. *Assume that the DLP is hard and h is a random oracle hash function. Then it is hard for the dishonest prover to construct a pair $< h(X), T >$ with $X = TV$.*

Proof. The dishonest prover knows

$$P, xP, C^x = \hat{e}(P, xP), W = C^r, h(V)$$

and he does not know r and x in Z_q^*. Thus we may assume that the dishonest prover computes $< h(X'), T' >$ with $X' = T'V$. Then $X' = C^{rx'+rz'x'}$, and $T'V = C^{rz'x'+rx'}$. From $X' = T'V$, we have $C^{rx'} = C^r x$. Since the mapping $f_P : G_1 \times G_1 \to G_2$ given by $f_P(Q) = \hat{e}(Q, P)$ is one-to-one mapping [4], we have

$$C^{rx} = C^{rx'} \Leftrightarrow \hat{e}(rxP, P) = \hat{e}(rx'P, P)$$
$$\Leftrightarrow f_P(rxP) = f_P(rx'P) \Leftrightarrow rxP = rx'P.$$

Let $R = rP$ and $Q = rxP$. Thus we know that to construct a pair $< h(X), T >$ with $X = TV$ for unknowns $r, x \in Z_q^*$ is to construct x' satisfying $x'R = Q$ for the known $R, Q \in G_1$. This is the Discrete-Logarithm Problem (DLP) and thus it is hard for a dishonest prover to construct $< h(X), T >$ with $X = TV$. □

4.2. Honest verifier zero-knowledge

The transcript consists of the messages exchanged between the two parties. The definition of perfect zero-knowledge can be found in [3]. In Theorem 4.2, 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 4.2. *Protocol 3.2 is perfect zero-knowledge for an honest verifier.*

Proof. Let D_r be the set of all real transcripts obtained by a prover and an honest verifier as the following form;

$$D_r = \{h(V), W, h(X), T\}$$
$$= \{h(C^{rx}, C^r, h(C^{rx+rzx}|z, x \in Z_q^*\}$$

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 E_r be the set of simulated transcripts can be constructed by the verifier

as following; The verifier chooses a uniformly at random from Z_q^* and computes the set E_r of the simulated transcripts by

$$E_r = \{h(C^{rx}), C^r, h(C^{rx+r\alpha}), C^{r\alpha} | \alpha \in Z_q^*\}$$

using $C^{rx} = \hat{e}(rxP, P), C^r = \hat{e}(rP, P)$ and $C^{rx+r\alpha} = \hat{e}(rxP, P)\hat{e}(r\alpha P, P)$. Then we have $D_r = E_r$. That is, D_r and E_r have identical probability distribution. Therefore the above protocol is perfect zero-knowledge for an honest verifier.

□

4.3. Dishonest verifier zero-knowledge

A dishonest verifier will generate $< h(V), W >$ with $h(V) = h(W^x)$ non-uniformly. In other 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 4.3. *Assume that the DLP is hard and h is a random oracle hash function. Then it is hard for a dishonest verifier to construct V such that $h(V) = h(W^x)$ for given W, P, and xP.*

Proof. To construct V, a dishonest verifier must construct $C^{r'x}$ such that $C^{r'x} = C^{rx}$ for unknowns $r, x \in Z_q^*$. By the same method as the proof of Theorem 4.1, $C^{r'x} = C^{rx}$ if and only if $r'xP = rxP$. Thus to construct V for a dishonest verifier is equivalent to compute $r' \in Z_q^*$ such that $r'Q = R$ for given $P, xP = Q$, $rxP = R$. This is the Discrete-Logarithm Problem and so it is hard. □

4.4. Passive adversary blindness

Our protocol has a passive adversary blindness property. That is, any polynomially bounded adversary has not a non-negligible advantage in deciding the honesty of the participants in the protocol. Assuming that the DLP is intractable, we have

1. It is impossible for a passive adversary to decide the honesty of the prover: given $P, xP, h(V), W$, deciding if $V = W^x$ is an instance of the DLP as Theorem 4.3.
2. Similarly it is impossible for a passive adversary to decide the honesty of the verifier: given $P, xP, h(X), T$ deciding if $X = TV$ is an instance of the DLP as Theorem 4.1.

4.5. Knowledge extractor

Let $L_1 = \{< h(X), T > | X = TV\}$. Then a prover ID essentially proves knowledge of the witness $< h(X), T > \in L_1$ using the shared string $< P, xP, C^x, rP, h(rxP) >$. Clearly $L_1 \in NP$. Assume that a dishonest prover ID^* is able to make any verifier accept. That is, given $< P, xP, C^x, rP, h(rxP) >$, ID^* can always output a pair $< h(X'), T' >$ such that $X' = T'V$. By simulating the honest verifier itself and by the hardness of DLP, ID^* can not obtain $< h(X'), T' >$, the witness that $< X', T' > \in L_1$. Thus our protocol is a "proof of knowledge"

5. Signature

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 5.1: Hidden signature protocol

1. Initialization : B asks A to identify itself by sending the challenge $< h(V), W >$ in the first step of Protocol 3.2.
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 W^x and check that $h(V) = h(W^x)$. And then A choose $z \in Z_q^*$ randomly and compute $h(X) = h(TV^\beta) = h(C^{r\beta x + rzx\beta})$ and $T = W^{rzx}$. The 3-pair $< h(X), T, M >$ is sent to B.
3. Verification : After receiving $< h(X), T, M >$, B extracts the signature $S = TV$. The verification condition is $h(X) = h(S^\beta)$.

6. Conclusion

In this paper, we proposed a new secure identification and signature protocol using zero-knowledge. Only based on the DLP assumption, it is secure in a random oracle model. Also in our protocol the only verifier uses bilinear pairings but not the prover. Thus smart cards with our scheme need not have devices for bilinear pairings. Under the methods of security proof given by Stinson and Wu [11], our protocol is secure against active-intruder attacks but Saxena et al.'s scheme [10] has a weakness of them. Also our proposed protocol is more simple and fast than YWL's scheme[9] because we reduced the number of computations.

References

[1] M. Bellare and P. Rogaway, *Entity authentication and key distribution*, Lecture Notes in computer Science **773** (1994), 232-149 (CRYPTO '93 Proceedings).

[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 (CRYPTO '86 Proceedings).

[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** (1991), no. 3, 690-728.

[9] Y. W. Lee, *Blind identification using bilinear pairings for smart cards*, J. Appl. Math. & Informatics, (2008), to appear.

[10] A. Saxena, B. Soh and S. Priymak, *Zero-Knowledge blind identification for smart cards using bilinear pairings*, Cryptology e-Print Archive, Report 2005/343, (2005).

[11] D.R. Stinson and J. Wu, *An efficient and secure two-flow zero-knowledge identification protocol*, Cryptology e-Print Archive, report 2006/337, 2006.

*

Department of Computer and Information Security,
Daejeon University,
300-716 Daejeon, Republic of Korea
E-mail: bmchoi@dju.ac.kr

**

Department of Computer and Information Security,
Daejeon University,
300-716 Daejeon, Republic of Korea
E-mail: ywlee@dju.ac.kr