JOURNAL OF THE
CHUNGCHEONG MATHEMATICAL SOCIETY
Volume **20**, No. 4, December 2007

ZERO-KNOWLEDGE GROUP IDENTIFICATION AND HIDDEN GROUP SIGNATURE FOR SMART CARDS USING BILINEAR PAIRINGS

Young Whan Lee* and Byung Mun Choi**

Abstract. In this paper, we propose a new blind group identification protocol and a hidden group signature protocol as its application. These protocols involve many provers and one verifier such that (1) the statement of all the provers are proved simultaneously, (2) and also all the provers using computationally limited devices (e.g. smart cards) have no need of computing the bilinear pairings, (3) but only the verifier uses the bilinear pairings. A. Saxena et al. proposed a two-round blind (group) identification protocol in 2005 using the bilinear pairings. But it reveals weakness in the active-intruder attack, and all the provers as well as the verifier must have devices computing bilinear pairings.

Comparing their results, our protocol is secure from the active-intruder attack and has more fit for smart cards. In particular, it is secure under only the assumption of the hardness of the Discrete-Logarithm Problem in bilinear groups.

1. Introduction

A zero-knowledge blind group identification scheme enables a group of users to identify themselves to a server such that (a) if all users are honest the server always accepts and (b) if any users are dishonest the server always rejects. However, in this case it is impossible to find out the actual identity of the particular cheating users.

For example, Alice and Bob want to identity themselves jointly to a server, and they don't trust each other to individually login to the

---

Received August 31, 2007.

2000 Mathematics Subject Classification: Primary 39B72, 39B22.

Key words and phrases: Identification, Signature, Smart card, Bilinear pairing.

*This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2006-521-D00475).

server without the other's approval. Alice wants to ensure that the identification succeeds if and only if the other user is really Bob. Bob has a similar requirement.

A. Saxena, B. Soh and S. Priymat [13] proposed a two-round blind (group) identification using bilinear pairings. But their protocol has a weakness of the active-intruder attack. Also in their protocol all the provers as well as the verifier need to compute bilinear pairings with some devices. But pairing implementation attempts in limited devices such as smart cards reveal that the embedded code may be slow, resource-consuming and tricky to program, although pairing is a cubic-time implementation [5].

To improve these two weaknesses, we propose a new zero-knowledge blind (group) identification protocol for smart cards. First, the bilinear pairings will be used only to verifier but not to the prover in our protocols for identifications and signatures. Secondly, our protocol is strong under the active-intruder attack and is secure assuming the hardness of the Discrete-Logarithm problem in bilinear groups. Also when a group of the provers identifies jointly to the server, they also send plain text messages with hidden signatures such that only the server can extract the signature.

The organization of paper is as follows. In Section 2, we present the preliminaries of bilinear parings and background, and give an example of the active-intruder attack on Saxena et al.'s blind group identification scheme. In Section 3 we propose our new two-round group identification and then in Section 4 we prove the security of the proposed protocol. In Section 5 we derive the hidden signature from our scheme. Finally, a conclusion is given in Section 6.

2. Bilinear pairings and background

2.1. Bilinear pairings

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

$\hat{e} : G_1 \times G_1 \rightarrow G_2$ 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 [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 $rP \in G_1$.

2.2. 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 [13]. Assume that $\{A_1, A_2, \cdots, A_n\}$ are the set of users who want to jointly identify themselves. It is necessary that each user A_i must have a certified public key $Y_i = x_i P_i$ where $P_i \in G_1$. The goal of the protocol is that all users will simultaneously identify themselves to the server S.

1. The SSP (A. Saxena, B. Soh and S. Priymak [13]) Blind Group Identification Scheme
   (1) The n provers $A_1, A_2, \cdots, A_n$ start by claiming to the server S that they know the discrete logarithms $x_1, x_2, \cdots, x_n \in Z_q^*$ of $A_1, A_2, \cdots, A_n \in G_1$ (to base P) respectively.
   (2) The verifier S generates $r_1, r_2, \cdots, r_n \in Z_q^*$ uniformly at random and compute $R_i = r_i Y_i$ and $U_i = r_i^2 P_i$. It makes the list of challenges $< A_i, R_i, U_i >$ public.
   (3) Each A_i computes $V_i = \frac{1}{x_i} R_i$ and checks that $\hat{e}(V_i, V_i) = \hat{e}(U_i, P)$; if the test passes, it generates $Q_i \in G_1$ and computes $Z_i = V_i + x_i Q_i$.
   (4) All users then collaborate to jointly compute the value $Z = \sum_{i=1}^{i=n} Z_i$. This computation is hidden from S so that individual values Z_i are effectively kept secret from its view. The combined

proof $< Z, Q_1, Q_2, \cdots, Q_n >$ is sent to S.
(5) S accepts if $\hat{e}(Z - \sum_{i=1}^{i=n} r_iP, P) = \prod_{i=1}^{i=n} \hat{e}(Q_i, Y_i)$

2. Active-intruder Attack on SSP Blind Group Identification Scheme
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 handle this definitional issue [1, 6, 15, 16]. 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 [16].

We give an example of active-intruder attack on SSP blind group identification scheme as follow: We use simple figures and notations to illustrate the SSP blind group identification protocol and corresponding active-intruder attacks on it. Let r_i be a random number chosen by the server S, x_i a random number chosen by provers $A_i (i = 1, 2, \cdots, n)$, and O any attacker. All computations take place in a relevant group.

SSP blind group identification scheme:
Note that x_i is secret key and x_iP_i is public key for each $A_i(i = 1, 2, \cdots, n)$

$$A_1 \xleftarrow{<R_1=r_1x_1P,U_1=r_1^2P>} B$$

$$A_i \xleftarrow{<R_i=r_ix_iP,U_i=r_i^2P>} B$$

$$\vdots$$

$$A_n \xleftarrow{<R_n=r_nx_nP,U_n=r_n^2P>} B$$

$$\{A_1, A_2, \cdots, A_n\} \xrightarrow{<Z=\sum_{i=1}^{i=n} Z_i,Q_1,Q_2,\cdots,Q_n>} S$$

A_i verifies that $\hat{e}(\frac{1}{x_i}R_i, \frac{1}{x_i}R_i) = \hat{e}(U_i, P)$. If the test passes, it generates $Q_i \in G_1$ and computes $Z_i = V_i + x_iQ_i$, where $V_i = \frac{1}{x_i}R_i$.
Also S verifies that $\hat{e}(Z - \sum_{i=1}^{i=n} r_iP, P) = \prod_{i=1}^{i=n} \hat{e}(Q_i, Y_i)$ and accepts.

Attack : The active-intruder attack is possible as follows :

$$A_1 \xleftarrow{<2R_1=2r_1x_1P,4U_1=4r_1^2P>} O \xleftarrow{<R_1=r_1x_1P,U_1=r_1^2P>} B$$

$$A_2 \xleftarrow{<2R_2=2r_2x_2P,4U_2=4r_2^2P>} O \xleftarrow{<R_2=r_2x_2P,U_2=r_2^2P>} B$$

$$\vdots$$

$$A_n \xleftarrow{<2R_n=2r_nx_nP,4U_n=4r_n^2P>} O \xleftarrow{<R_n=r_nx_nP,U_n=r_n^2P>} B$$

$$\{A_1, A_2, \cdots, A_n\} \xrightarrow{<Z=\sum_{i=1}^{i=n} Z_i, Q_1, Q_2, \cdots, Q_n>} O \xrightarrow{<\frac{1}{2}Z, \frac{1}{2}Q_1, \frac{1}{2}Q_2, \cdots, \frac{1}{2}Q_n>} S$$

A_i verifies that

$$\hat{e}(\frac{1}{x_i}2R_i, \frac{1}{x_i}2R_i) = \hat{e}(2r_iP, 2r_iP) = \hat{e}(P,P)^{4r_i^2} = \hat{e}(4r_i^2P, P) = \hat{e}(4U_i, P).$$

If the test passes, it generates $Q_i \in G_1$ and computes $z_i = V_i + x_iQ_i$, where $V_i = \frac{1}{x_i}R_i$. S verifies that

$$\hat{e}(\frac{1}{2}Z - \sum_{i=1}^{i=n} r_iP, P) = \prod_{i=1}^{i=n} \hat{e}(Q_i, P)^{\frac{x_i}{2}} = \prod_{i=1}^{i=n} \hat{e}(\frac{1}{2}Q_i, Y_i) = \hat{e}(\frac{1}{2}Q_i, xP)$$

and accepts.

2.3. Our contribution

In this paper, we propose a new blind group identification protocol for smart cards using a public key cryptosystem. Our protocol has several advantages.

1. Every prover with computationally limited device such as smart cards does not use bilinear pairings and only the server uses them.
2. Our protocol is secure assuming only the hardness of the Discrete-Logarithm Problem in bilinear groups. Note that the SSP blind group identification scheme and the SW (D. R. Stinson and J. Wu) identification scheme need another assumption such as the hardness of the DHP, EDHP or LDHP [13, 16].
3. The SSP blind group identification scheme has a weakness of the active-intruder attack, but our scheme does not.
4. Our protocol devices the hidden group signature.

3. Our new blind identification

3.1. Setup PKI(Public Key Infrastructure)

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 3.1: Group 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_i chooses a private key x_i uniformly from Z_q^* at random, computes x_iP and registers x_iP as A_i's public key for each $i = 1, 2, \cdots, n$.

3.2. Group identification protocol description

This scheme enables a group of provers (users) to identify themselves to a verifier (server) such that: (a) The identification test passes if none of the provers cheat, (b) if any of the provers cheat, the test will fail with a high prob-ability, (c) it is not possible for the verifier or the provers to know who cheat. The steps in a session of our scheme as follows:

Protocol 3.2: A group identification scheme

Let $\{A_1, A_2, \cdots, A_n\}$ be the set of provers who want to identify themselves. It is necessary that each prover A_i must have a certified public key $Y_i = x_iP$ as Protocol 3.1. The goal of the scheme is that all provers will simultaneously identify themselves to a verifier S. That is, the proof is valid only on all the statements together: "A_i knows x_i" for all $i = 1, 2, \cdots, n$ but not on any of the individual statements like "A_1 knows x_1" or "A_2 knows x_2" independently of the others. We will assume the infrastructure of Protocol 3.1. The identification is done as follows:

1. The verifier S chooses $r_1, r_2 \cdots, r_n \in Z_q^*$ uniformly at random, and computes $V_i = \hat{e}(r_ix_iP, x_iP) = C^{r_ix_i^2}$, $W_i = \hat{e}(r_iP, x_iP) = C^{r_ix_i}$ and $h(V_i)$. Then S sends $< h(V_i), W_i >$ to the prover A_i for each $i = 1, 2, \cdots, n$.
2. After receiving $< h(V_i), W_i >$, A_i rejects and stops if $h(V_i) \neq h(W_i^{x_i})$, or $W_i \notin G_2$; otherwise A_i chooses $z_i \in Z_q$, and compute $X_i = W_i^{\frac{1}{x_i}} C^{x_i^3 z_i}$ and $T_i = V_i^{x_i z_i} = W_i^{x_i^2 z_i}$ for each $i = 1, 2, \cdots, n$. All provers then collaborate to jointly compute the value $X = \prod_{i=1}^{i=n} X_i$. This computation is hidden from S so that individual values $< X_i, T_i >$ are effectively kept secret from its view. The combined proof $< X, T_1, T_2, \cdots, T_n >$ is sent to S.

3. After receiving $< X, T_1, T_2, \cdots, T_n >$, S accepts if $X = \prod_{i=1}^{i=n} C^{r_i} T^{\frac{1}{r_i}}$, otherwise S rejects.

3.3. Completeness of Protocol 3.2

It is straightforward to prove that Protocol 3.2 is complete. Suppose $\{A_1, A_2, \cdots, A_n\}$ and S are all honest. After receiving the challenge $< h(V_i), W_i >$ for each $i = 1, 2, \cdots, n$, A_i checks to see if $h(V_i) = h(W_i^{x_i})$. Since $V_i = C^{r_i x_i^2} = (C^{r_i x_i})^{x_i} = W_i^{x_i}$ for each $i = 1, 2, \cdots, n$, A_i accepts and all provers A_i then collaborate to jointly compute the value $X = \prod_{i=1}^{i=n} X_i$. The combined proof $< X, T_1, T_2, \cdots, T_n >$ is sent to S. Then S checks to see if $X = \prod_{i=1}^{i=n} C^{r_i} T_i^{\frac{1}{r_i}}$. Since

$$X = \prod_{i=1}^{i=n} X_i = \prod_{i=1}^{i=n} W_i \frac{1}{x_i} C^{x_i^3 z_i} = \prod_{i=1}^{i=n} C^{r_i} (C^{r_i x_i^3 z_i})^{\frac{1}{r_i}} = \prod_{i=1}^{i=n} C^{r_i} T^{\frac{1}{r_i}},$$

S also accepts.

4. Security of the proposed group identification protocol

In this section, we prove that the above protocol is perfect zero-knowledge using the restricted definition of Bounded-prover perfect Zero-knowledge (BP-pZK)[3], which essentially requires that the probability of the dishonest verifier succeeding is negligibly less than that of a dishonest prover succeeding.

4.1. Soundness

Assuming an honest verifier, we must show that a dishonest prover cannot succeed except with a negligible probability. Given $x_i P$, $h(V_i)$, W_i for each $i = 1, 2, \cdots, n$, the task of a dishonest prover is to compute a pair $< X_i, T_i >$ such that $X_i = C^{r_i} T^{\frac{1}{r_i}}$. We show that this is an instance of the DLP in Theorem 1. The knowledge of W_i and $h(V_i)$ does not give a dishonest prover any additional advantage in solving this DLP instance because deciding if $h(V_i) \equiv h(W_i^{x_i})$ is an instance of the DLP as Theorem 3. 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. Then it is hard for the dishonest prover to construct a pair $< X_i, T_i >$ without knowledge of x_i for some $i (1 \leq i \leq n)$ such that $X = \prod_{i=1}^{i=n} X_i = \prod_{i=1}^{i=n} C^{r_i} T_i^{\frac{1}{r_i}}$.*

Proof. The dishonest knows

$$P, x_iP, C^{x_i} = \hat{e}(P, x_iP), C^{x_i^2} = \hat{e}(x_iP, x_iP), W_i = C^{r_ix_i}, h(V_i)$$

for each $i = 1, 2, \cdots n$ and he does not know r_i and x_i in Z_q^* for each $i = 1, 2, \cdots n$. Thus we may assume that $X_i = (C^{r_ix_i})^{\frac{1}{x_i'}}(C^{r_ix_i})^{x_i'^2 z_i}$ and $T_i = (C^{r_ix_i})^{x_i'^2 z_i}$ for some $x_i', z_i \in Z_q^*$, and $X_j = C^{r_j + x_j^3 z_j}$ and $T_j = C^{r_j + x_j^3 z_j}$ for all $j \neq i$. If $X = \prod_{i=1}^{i=n} X_i = \prod_{i=1}^{i=n} C^{r_i} T_i^{\frac{1}{r_i}}$, then we have $C^{r_i \cdot x_i \frac{1}{x_i'} + r_i x_i x_i'^2 z_i} = C^{r_i + r_i x_i x_i'^2 z_i}$. Let $f_P : G_1 \times G_1 \to G_2$ be the one-to-one mapping given by $f_P(Q) = \hat{e}(Q, P)$ [3]. Then we have

$$\begin{aligned} C^{r_ix_i} = C^{r_ix_i'} &\Leftrightarrow \hat{e}(r_ix_iP, P) = \hat{e}(r_ix_i'P, P) \\ &\Leftrightarrow f_P(r_ix_iP) = f_P(r_ix_i'P) \Leftrightarrow r_ix_iP = r_ix_i'P. \end{aligned}$$

That is, $r_ix_iP = r_ix_i'P$. Let $R = r_iP$ and $Q = r_ix_iP$. Thus we know that to construct a pair $< X_i, T_i >$ with $X = \prod_{i=1}^{i=n} X_i = \prod_{i=1}^{i=n} C^{r_i} T_i^{\frac{1}{r_i}}$ for unknowns $r_i, x_i \in Z_q^*$ is to construct x_i' satisfying $x_i'R = Q$ for the known $R, Q \in G_1$. This is the Discrete-Logarithm Problem and thus it is hard for a dishonest prover to construct $< X_i, T_i >$ with $X = \prod_{i=1}^{i=n} X_i = \prod_{i=1}^{i=n} C^{r_i} T_i^{\frac{1}{r_i}}$. □

4.2. Honest verifier zero-knowledge

The transcript consists of the messages exchanged between the two parties. In Theorem 2, we construct a simulator that can generate an accepting transcript $\{h(V_i), W_i, X_i, T_i, X\}$ 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. The set $\Im$ of real transcripts obtained by provers and an honest verifier consists of all transcripts $\Im$ having the following form:

$$\begin{aligned} \Im &=< h(V_i), W_i, X_i, T_i.X > \\ &=< h(C^{r_ix_i^2}), C^{r_ix_i}, C^{r_i + x_i^3 z_i}, C^{rx^3 z_i}, \sum_{i=1}^{i=n} X_i > . \end{aligned}$$

Note that r_i is chosen by the verifier uniformly at random from Z_q^* and also z_i is chosen by the prover uniformly at random from Z_q^*.

The set $\Im$ of simulated transcripts can be constructed by the verifier as follows. The verifier chooses r_i and α_i uniformly at random from Z_q^* and using $h(\hat{e}(r_i x_i P, x_i P)), \hat{e}(r_i P, x_i P), \hat{e}((r_i + \alpha_i)P, P), \hat{e}(r_i \alpha_i P, P)$ and $\prod_{i=1}^{i=n} \hat{e}((r_i \alpha_i)P, P)$ computes the simulated transcript

$$\hat{\Im} = \{h(C^{r_i x_i^2}), C^{r_i x_i}, C^{r_i + \alpha_i}, C^{r_i \alpha_i}, \prod_{i=1}^{i=n} C^{r_i + \alpha_i}\}.$$

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

4.3. Dishonest verifier zero-knowledge

A dishonest verifier will generate $< V_i, W_i >$ with $h(V_i) = h(W_i^{x_i})$ non-uniformly for some $i (1 \leq i \leq n)$. In order words, a dishonest verifier will not know r_i corresponding to V_i for some $i (1 \leq i \leq n)$. 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 $h(\cdot)$ is random oracle. Then it is hard for a dishonest verifier to construct W_i such that $h(V_i) = h(W_i^{x_i})$ for given $V_i, P, x_i P (i \in \{1, 2, \cdots, n\})$.*

Proof. To construct W_i, a dishonest verifier must find r_i' such that $C^{r_i' x_i^2} = C^{r_i x_i^2}$ for unknowns $r_i, x_i \in Z_q^*$. Let $f_{x_i P} : G_1 \times G_1 \to G_2$ be the one-to-one mapping given by $f_{x_i P}(Q) = \hat{e}(Q, x_i P)$ [3]. Then we have

$$C^{r_i' x_i^2} = C^{r_i x_i^2} \Leftrightarrow \hat{e}(r_i' x_i^2 P, P) = \hat{e}(r_i x_i^2 P, P)$$
$$\Leftrightarrow f_{x_i P}(r_i' x_i P) = f_{x_i P}(r_i x_i P) \Leftrightarrow r_i' x_i P = r_i x_i P.$$

Thus to construct W_i is equivalent that given $P, x_i P = Q$, $r_i x_i P = R$ and unknowns $r_i, x_i \in Z_q^*$ a dishonest verifier compute r_i' such that $r_i' Q = R$. This is the Discrete-Logarithm Problem and so it is hard. □

4.4. 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: for any $i = 1, 2, \cdots, n$ and given $P, x_i P, C^{x_i}, C^{x_i^2}, W_i, h(V_i)$, deciding if $V_i = W_i^{x_i}$ is an instance of the DLP. Similarly it is impossible for a passive adversary to decide the honesty of the prover: given $P, x_iP, C^{x_i}, C^{x_i^2}, W_i, h(V_i)X_i, T_i$, for any $i = 1, 2, \cdots, n$, deciding if $X = \prod_{i=1}^{i=n} X_i = \prod_{i=1}^{i=n} C^{r_i} T_i^{\frac{1}{r_i}}$ is an instance of the DLP.

4.5. Knowledge extractor

Let $L_i = \{< X_i, T_i > | X_i = C^{r_i} T_i^{\frac{1}{r_i}}\}$ for any $i = 1, 2, \cdots, n$. Then a prover A_i essentially proves knowledge of the witness $< X_i, T_i > \in L_i$ using the shared string $< P, x_iP, C^{x_i}, C^{x_i^2}, C^{r_i x_i}, h(C^{r_i x_i^2}) >$ for all $i = 1, 2, \cdots, n$. Clearly $L_i \in NP$ for all $i = 1, 2, \cdots, n$.

Assume that a dishonest prover A_i^* is able to make any verifier accept. That is, given $< P, x_iP, C^{x_i}, C^{x_i^2}, C^{r_i x_i}, h(C^{r_i x_i^2}) >$, A_i^* can always output a pair $< X_i', T_i' >$ such that $X' = \prod_{i=1}^{i=n} X_i' = \prod_{i=1}^{i=n} C^{r_i} T_i'^{\frac{1}{r_i}}$. By simulating the honest verifier itself, A^* can obtain $< X_i', T_i' >$, the witness that $< X_i', T_i' > \in L_i$ for each $i = 1, 2, \cdots, n$. Thus our protocol is a "proof of knowledge"

5. Hidden group signatures

In this section we provide a hidden group signature scheme. All users $\{A_1, A_2, \cdots, A_n\}$ can also jointly send plain text message along with hidden group signature such that S can extract the signature.

Protocol 5.1: Hidden group signature scheme

1. Initialization : S asks A_i for all $i = 1, 2, \cdots, n$ to identify itself by sending the challenge $< h(V_i), W_i >$ in the first step of Protocol 3.2.

2. Signing : Let $M \in G_1$ be the message to be signed and $H(M) = w$, where $H : G_1 \to Z_q^*$ is a hash function. For each $i = 1, 2, \cdots, n$, A_i computes $W_i^{x_i}$ and check that $h(V_i) = h(W_i^{x_i})$. And then A_i choose $z_i \in Z_q^*$ randomly and compute $X_i = W_i^{\frac{w}{x_i}} C^{z_i w x_i^3}$ and $T_i = W_i^{x_i^2 z_i}$ for all $i = 1, 2, \cdots, n$.
3. All provers then collaborate to jointly compute the value $X = \prod_{i=1}^{i=n} X_i$. This computation is hidden from S so that individual values $< X_i, T_i >$ are effectively kept secret from its view. The combined proof $<< X, T_1, T_2, \cdots, T_n >, M >$ is sent to S.

4. Verification : After receiving $<< X, T_1, T_2, \cdots, T_n >, M >$, S extracts the signature $Sig(M) = \prod_{i=1}^{i=n} C^{r_i} T_i^{\frac{1}{r_i}}$. The verification condition is $X = Sig(M)^w$.

6. Conclusion

In this paper, we proposed a new zero-knowledge blind group identification protocol for smart cards. Only with the DLP assumption, it is secure in random oracle model. Also in our protocol the only verifier uses bilinear pairings but not the provers. Thus smart cards with our scheme need not have devices for bilinear pairings. Under the methods of security proof given by Stinson and Wu [16], our protocol is secure against the active-intruder attacks but Saxena et al.' scheme [13] has a weakness of them.

References

[1] M. Bellare and P. Rogaway, *Entity authentication and key distribution*, Lec-ture 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**:390-420, 1993.

[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 Cryptol-ogy Conference on Advances in Cryptology, Springer-Verlag, (2001), 213-229.

[5] B. Chevallier-Mames, J. S. Coron, N. McCullagh, D. Naccache and M. Scott, *Secure delegation of elliptic curve pairing*, Cryptology e-Print Archive, report 2005/150, (2005).

[6] W. Diffie, P.C. van Oorschot and M.J. Wiener, *Authentication and Authenticated key exchanges, Designs, Codes and Cryptography 2* , (1992), 107-125

[7] U. Feige, A. Fiat, and A. Shamir, *Zero knowledge proofs of identity*, J. Cryptology 1 (1988), 77-94.

[8] 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).

[9] 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

[10] L. Guillou and J.J. Quisquater, *A "paradoxical" identity-based signature scheme re-sulting from zero-knowledge*, Lecture Notes in computer Science 403 (1990), 216-231 (CRYPTO '88 Proceedings).

[11] A.J. Menezes, P.C. van Oorschot and S.A. Vanstone, *Handbook of Ap-plied Cryptography*, CRC Press, 1996.
[12] T. Okamoto, *Provably secure and practical identification schemes and corresponding signature schemes*, Lecture Notes in Computer Science 740 (1993), 31-53 (CRYPTO '92 Proceedings).
[13] 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.
[14] C.P. Schnorr, *Efficient signature generation by smart cards*, Journal of Cryptology, **4** (1991), 161-174
[15] D.R. Stinson, *Cryptography: Theory and Practice, Third Edition*, Chapman & Hall/CRC, Boca Rayon, 2006.
[16] 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,
Daejeon, 300-716, Republic of Korea
E-mail: ywlee@dju.ac.kr

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