J. Appl. Math. & Informatics Vol. **26***(2008), No. 5 - 6, pp. 1139 - 1147*
Website: http://www.kcam.biz

BLIND IDENTIFICATION USING BILINEAR PAIRINGS FOR SMART CARDS

YOUNG WHAN LEE

ABSTRACT. 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. In this paper, we give a method of the active-intruder attack on their identification scheme and propose a new zero-knowledge blind identification protocol for smart cards. Our protocol consists of only two message flows and does not rely on any underlying signature or encryption scheme. The prover using computationally limited devices such as smart cards has no need of computing the bilinear pairings. It needs only for the verifier. 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, Smart card, Bilinear pairing.

1. Introduction

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 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. There are many identification protocols using zero-knowledge proofs [2, 6, 7, 8]. A. Saxena et al.[9] 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

Received January 17, 2008. Accepted June 4, 2008.

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

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. Also their scheme has a weakness of the active-intruder attack. To improve these weaknesses, we propose a new zero-knowledge blind identification protocol for smart cards. The bilinear pairings be used only to verifier but not prover in our protocols of 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 the preliminaries of bilinear parings and background, and give a method of the active intruder attack on Saxena et al.'s scheme. In Section 3 we propose our new two-round blind identification and then in Section 4 we prove the security of the proposed protocol. In Section 5 we deal with other extensions such as hidden signatures and anonymous seller credit payment. 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, 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 [9]. 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 [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 [9]) 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 $\hat{e}\left(\frac{1}{x}R, \frac{1}{x}R\right) \neq \hat{e}(U, P)$; 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) Active-intruder Attack on SSP 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 handling this definitional issue [1, 5, 10]. 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 [10]. We give an example of the active-intruder attack on SSP scheme as follow:

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 $\hat{e}\left(\frac{1}{x}R, \frac{1}{x}R\right) \equiv \hat{e}(U, P)$ and accepts. Also B verifies that 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

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

and accepts. B verifies that

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

and accepts.

2.3. 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 in our protocol does not use bilinear pairings and only the verifier uses them.

(2) Our protocol is secure assuming only the hardness of the Discrete Logarithm Problem in bilinear groups. Note that the SSP scheme needs another assumption such as the hardness of the DHP, EDHP or LDHP [9].

(3) The SSP scheme has a weakness of the active-intruder attack, but our scheme does not.

3. Our new two-round blind identification

3.1. 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 3.1: 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 $\hat{e} : G_1 \times G_2 \rightarrow G_2$.

(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 $h : G_1 \rightarrow \{0,1\}^k$.

(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 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 3.2: A 2-flow identification scheme

(1) The verifier B chooses $r \in Z_q^*$ uniformly at random, and computes $V = \hat{e}(rxP, xP) = C^{rx^2}$, $W = \hat{e}(rP, xP) = C^{rx}$ 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 $X = W^{\frac{1}{x}} C^{x^3 z}$ and $T = W^{x^2 z}$. Then A sends $< X, T >$ to B.

(3) After receiving $< X, T >$, B accepts if $X = C^r T^{\frac{1}{r}}$; otherwise B rejects.

3.3. Completeness

It is straightforward to prove that Protocol 3.2 is complete. Suppose A and B are both honest. After receiving the challenge $< h(V), W >$, A checks to see if $h(V) \equiv h(W^x)$. Since $V = C^{rx^2} = (C^{rx})^x = W^x$, A accepts and sends the response $< X, T >$ to B. Then B checks to see if $X = C^r T^{\frac{1}{r}}$. Since

$$X = W^{\frac{1}{x}} C^{x^3 z} = C^r (C^{rx^3 z})^{\frac{1}{r}} = C^r T^{\frac{1}{r}},$$

B also accepts.

4. Security of the proposed protocol

In this section, we prove that the above 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, $h(V)$, W the task of a dishonest prover is to compute a pair $< X, T >$ such that $X = C^r T^{\frac{1}{r}}$. We show that this is an instance of the DLP in 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 $h(V) = h(W^x)$ is an instance of the DLP as Theorem 4.3. Thus, the proof is sound from a verifier's view as long as the DLP is intractable.

Theorem 1. *Assume that the DLP is hard. Then it is hard for the dishonest prover to construct a pair* $< X, T >$ *with* $X = C^r T^{\frac{1}{r}}$.

Proof. The dishonest knows

$$P,\ xP,\ C^x = \hat{e}(P, xP),\ C^{x^2} = \hat{e}(xP, xP), W = C^{rx} \text{ and } h(V)$$

and he does not know r and x in Z_q^*. Thus we may assume that

$$X = (C^{rx})^{\frac{1}{x'}}(C^{rx})^{x'^2 z} \text{ and } T = (C^{rx})^{x'^2 z}$$

for $x', z \in Z_q^*$. If $X = C^r T^{\frac{1}{r}}$, then $C^{r \cdot x \frac{1}{x'} + xx'^2 z}$. Let $f_P : G_1 \times G_1 \to G_2$ be the one-to-one mapping given by $f_P(Q) = \hat{e}(Q, P)$ [4]. Then 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 $< X, T >$ with $X = C^r T^{\frac{1}{r}}$ 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 and thus it is hard for a dishonest prover to construct $< X, T >$ with $X = C^r T^{\frac{1}{r}}$. □

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

$$\{h(V), W, X, T\}$$

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 2. *Protocol 3.2 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 $\Im$ having the following form:

$$\Im =< h(V), W, X, T >=< h(C^{rx^2}), C^{rx}, C^{r+x^2 z}, C^{rx^3 z} >$$

where r is chosen by the verifier uniformly at random from Z_q^* and also z 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 and α uniformly at random from Z_q^* and computes the simulated transcript $\Im =< h(C^{rx^2}), C^{rx}, C^{r+\alpha}, C^{r\alpha} >$ using

$$h(\hat{e}(rxP, xP)), \hat{e}(rP, xP), \hat{e}((r + \alpha)P, P)$$

and $\hat{e}(r\alpha P, P)$. Since the random numbers r, z and α in Z_q^* have identical probability distributions and $\alpha = x^3(x^{-3}z)$, $\Im$ 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 $< V, W >$ with $h(V) = h(W^x)$ 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 3. *Assume that the DLP is hard and $h(\cdot)$ is random oracle. Then it is hard for a dishonest verifier to construct W such that $h(V) = h(W^x)$ for given V, P, xP.*

Proof. To construct W, a dishonest verifier must construct $C^{r'x^2}$ such that $C^{r'x^2} = C^{rx^2}$ for unknowns $r, x \in Z_q^*$. Let $f_{xP} : G_1 \times G_1 \to G_2$ be the one-to-one mapping given by $f_P(Q) = \hat{e}(Q, P)$ [4]. Then we have

$$C^{r'x^2} = C^{rx^2} \Leftrightarrow \hat{e}(r'x^2P, P) = \hat{e}(rx^2P, P)$$
$$\Leftrightarrow f_{xP}(r'xP) = f_{xP}(rxP) \Leftrightarrow r'xP = rxP.$$

Thus to construct W is equivalent that given $P, xP = Q, rxP = R$ and unknowns $r, x \in Z_q^*$ a dishonest verifier compute r' such that $r'Q = R$. 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: given $P, xP, C^x, C^{x^2}, W, h(V)$, deciding if $V = W^x$ is an instance of the DLP. Similarly it is impossible for a passive adversary to decide the honesty of the prover: given $P, xP, C^x, C^{x^2}, W, h(V), X, T$, deciding if $X = C^rT^{\frac{1}{r}}$ is an instance of the DLP.

4.5. Knowledge extractor

Let $L_1 = \left\{ < X, T > | X = C^rT^{\frac{1}{r}} \right\}$. Then a prover ID essentially proves knowledge of the witness $< X, T > \in L_1$ using the shared string

$$< P, xP, C^x, C^{x^2}, C^{rx}, h(C^{rx^2}) > .$$

Clearly $L_1 \in NP$. Assume that a dishonest prover ID^* is able to make any verifier accept. That is, given $< P, xP, C^x, C^{x^2}, C^{rx}, h(C^{rx^2}) >$, ID^* can always output a pair $< X', T' >$ such that $X' = C^r T'^{\frac{1}{r}}$. By simulating the honest verifier itself, ID^* can obtain $< X', T' >$, the witness that $< X', T' > \in L_1$. Thus our protocol is a "proof of knowledge"

5. Other Extensions

In this section we provide extensions of our scheme. The private keys $x_i \in Z_q^*$ can either be generated by the user or by a trusted authority. The public key $x_i P$ are assumed to be certified in the former case. A. Saxena et al. [9] introduced how to use their scheme for smart cards as follows;

The private key for each smart card is encapsulated in a tamper proof chip. Signing access to this key is given via some mechanism like a PIN number. The corresponding public key is also present in the smart card along with a certificate. Smart cards may be purchased from a (reputed) third party and must be registered with the relevant authority (like a bank) before they can be used. To register a smart card, the authority simple provide a certificate.

In our scheme smart cards do not need to have the device of the bilinear pairing.

5.1 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 5.1 : Hidden signature scheme

(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 $H(M) = w$, 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 $X = W^{\frac{w}{x}} C^{zwx^3}$ and $T = W^{x^2 z}$. The pair $<< X, T >, M >$ is sent to B.

(3) Verification : After receiving $<< X, T >, M >$, B extracts the signature $S = C^r T^{\frac{1}{r}}$. The verification condition is $X = S^w$.

5.2. Anonymous seller credit payments

We propose a secure blind identification and hidden signature scheme. Thus, as A. Saxena et al. proposed the secure of a anonymous seller credit payments, our scheme can also applies to them without the active-intruder attack as in the before.

6. 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 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 [10], our protocol is secure against active-intruder attacks but Saxena et al.'scheme [9] 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.
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 ASI-ACRYPT '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. 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).
10. D.R. Stinson and J. Wu, *An efficient and secure two-flow zero-knowledge identification protocol*, Cryptology e-Print Archive, Report 2006/337, (2006).

Young Whan Lee received BS and Ph.D at Chungnam Natioal University. Also he receved Ph.D in the department of computer enginneering at Myoungji University under the direction of Seoung Un Choi in 2003. He has been working as a faculty since 1988 at Daejeon University. His research interests focus on the functional analysis, cryptography and software engineering.

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