Privacy Enhancing Credentials

Junji Nakazato, Lihua Wang, and Akihiro Yamamura

National Institute of Information and Communications Technology
4-2-1, Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan
{nakazato, wlh, aki}@nict.go.jp

Abstract. Using pairing techniques, we propose an anonymous authenticated key exchange scheme based on credentials issued by a trusted third party. The protocol satisfies several security properties related to user privacy such as unforgeability, limitability, non-transferability, and unlinkability.

1 Introduction

Privacy issues have arisen because many users are becoming increasingly more concerned about how their sensitive information will be used. Current technology in electronic services such as e-commerce, e-business, and e-government allows service providers to easily track individual's actions, behaviors, and habits. The information (or data) obtained in the transactions may be sold for business purposes even though the users may not want this to be done. Therefore, they would like to conceal personal information related to their identity as much as possible. In this paper, we propose a new technique to solve privacy issues arising from open-network transactions. Such a scheme could have many applications. For example, potential applications could be for users to secure video rentals on demand or use single sign-on tickets. Video-on-demand systems allow them to select and watch videos over a network possibly as part of an interactive television system. For users to use such a systems, an access control should be implemented by service providers to verify their eligibility to watch these videos or request services. Thus, their identity may be exposed when a request reaches the service provider. Also, videos that users choose reflect taste and characteristics, matters that they want to keep confidential. Likewise, tickets to request service providers to provide services using a single sign-on scheme could leak user histories of what services they have requested. In either cases, users are exposed to threats where their sensitive information may be disclosed to undesirable entities.

There are two possible solutions. The first is to use oblivious transfer or private information retrieval to conceal which video a user has asked to watch. He/She can covertly ask the service provider to provide his/her request; however, this method does not address privacy issues in single sign-on schemes. The second is anonymize the request made by the user to the service provider. In this scenario, as the request received by the service provider is anonymous, the service provider may not necessarily have the means to determine whether the request is valid because the user cannot be traced back using the transmitted

I. Cervesato (Ed.): ASIAN 2007, LNCS 4846, pp. 55–61, 2007.

Table 1. Functions and requirements

		previous [4]	proposed
Functions	Credential Systems	Yes	Yes
	Flexibility of content[1]	Yes	No
	Authenticated key exchange	No	Yes
Requirements	Unforgeability	Yes	Yes
	Limitability	Yes	Yes
	Non-transferability[2]	No	Yes
	Anonymity	No	Yes
	Unlinkability	No	Yes

[1] Ng, Susilo, and Mu's scheme [4] can include any data in the credential, but our proposed protocol only contains data to share the key.

[2] Non-transferability means no party can transfer valid data to another. Consequently, it is different from Ng, Susilo, and Mu's [4].

request. This can be accomplished if a trusted third party issues a valid credential that enables the user to obtain the service from the provider anonymously without disclosing his/her identity. An anonymous credential system is an effective solution that can satisfy these properties. Organizations issue credentials to users for different organizations. Each organization knows user only by different pseudonyms respectively. Users can convince different organizations of only the fact that they have such credentials without revealing any information of the users (*anonymity*). Moreover, even if a user uses such a credential of multiple times, it cannot be linked to each other (*unlinkability*).

We report our current study of a scheme where a user and a service provider can establish an authenticated and secure channel after the protocol using a privacy enhanced credential in such a way that the scheme attains anonymity, unlinkability, unforgeability, limitability, and non-transferability.

The efficient anonymous credential proposed by J. Camenisch and A. Lysyanskaya is based on strong RSA assumption and Decision Diffie-Hellman (DDH) assumptions [2]. Our proposed scheme is similar to the one introduced by Ng, Susilo, and Mu [4]. Our proposed scheme achieves non-transferable anonymous credentials using a pairing technique on the elliptic curves over finite fields. Table 1 compares the functionalities and the security properties of their scheme with ours. We found that Ng, Susilo, and Mu's scheme could be tailored to fit such security requirements using different methods.

2 Proposed Scheme

2.1 Description of Proposed Scheme

We define participants in the proposed scheme as: an *authority*, a *user*, a *server* (or a *signer*), and *verifiers* (or *services*) satisfying the following properties:

Authority (A)**:** provides system parameters and public/private key pairs. It distributes these securely to all participants in the protocol. After that, A will not take part in the protocol.

User (U): wants to receive services from two or more specific V_j's. First, U sends a request to S, and receives a credential through a secure channel. When U wants to obtain a service, she/he sends a ticket that is generated from the valid credential designated to V_j by S.

Server (S): issues a credential to U for her/him to use the service provided by the designated V_j. S is a trusted third party.

Verifiers (V_j $(j = 1, \dots, n)$): check whether or not the ticket is valid. If so, V_j performs the protocol to exchange keys with U to establish an authenticated and secure channel, otherwise V_j does not.

We assumed the protocol flow between U and A would be carried out through secure channels and no adversaries could obtain any information. However, the protocol is carried out using an insecure channel when U sends a ticket to any V_j because the user is anonymous to V_j and therefore an authenticated channel cannot be employed for this purpose. Thus, any adversary can obtain information at this stage of the protocol.

2.2 Security Requirements

A malicious user may try to access to V_j, i.e., gain access that is not allowed. V_j should be able to detect these invalid attempts at access. At the same time, the protocol must protect the privacy of the user, even if V_j colludes with other V_i's. Therefore, our proposed scheme must at least satisfy *unforgeability*, *limitability*, *non-transferability*, and *unlinkability*.

Unforgeability: Nobody can forge a valid credential to generate a valid ticket with V_j without collaboration with S.

Limitability: U who was issued a valid credential by S can generate a ticket to V_j that is designated by S in the credential; U cannot forge a valid credential to generate a ticket to V_i that is not designated by S, even if he/she has been given some legitimate credential for V_j.

Non-transferability: There are two cases that should be considered:

1. V_j who received the ticket from U cannot forge it a ticket to a ticket to any other V_i.
2. U who was issued a valid credential by S cannot transfer it to any other U' to generate a valid ticket without leaking U's secret keys.

Unlinkability: U may use a credential issued by S to generate tickets to several verifier V_j's. No one can determine whether two tickets σ_1 and σ_2 have been generated by U. Even if V_1 colludes with V_2, no efficient algorithm exists to find the correlation between tickets sent to V_1 and V_2.

2.3 Bilinear Pairings and Complexity Assumption

The proposed scheme is based on *pairings*. A pairing is derived from either a modified Weil or Tate pairing on a supersingular elliptic curve or an abelian variety over a finite field (see [1,3] for further details). Let us briefly review the terminology and symbols that are used in the proposed scheme.

Let $\mathbb{G}_1$ denote an additive group of some large prime order q and $\mathbb{G}_2$ denote a multiplicative group also of order q. Let P denote a generator of $\mathbb{G}_1$. A map, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, is said to be an admissible bilinear pairing if the following properties hold:

1. Bilinear: Given any $Q, R \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q$, we have $\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$.
2. Non-degenerate: $\hat{e}(P, P) \neq 1_{\mathbb{G}_2}$.
3. Computable: There is an efficient algorithm to compute $\hat{e}(Q, R)$ for any $Q, R \in \mathbb{G}_1$.

The following three problems have been assumed to be intractable for any polynomial time algorithm.

Discrete Logarithm Problem: Given $P, aP \in \mathbb{G}_1$, find $a \in \mathbb{Z}_q^*$.

Computational Diffie-Hellman (CDH) Problem [1]: Given $P, aP, bP \in \mathbb{G}_1$, find $abP \in \mathbb{G}_1$.

Bilinear Diffie-Hellman (BDH) Problem [1]: Given $P, aP, bP, cP \in \mathbb{G}_1$, find $\hat{e}(P, P)^{abc} \in \mathbb{G}_2$.

2.4 Privacy Enhancing Designated Credentials

System parameters $params = (\mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, P, Q, F, \mathsf{H}(\cdot))$. P, Q, and F are non-trivial elements of $\mathbb{G}_1$ and let $\mathsf{H}(\cdot)$ be a hash function of $\{0,1\}^* \rightarrow \mathbb{G}_1$.

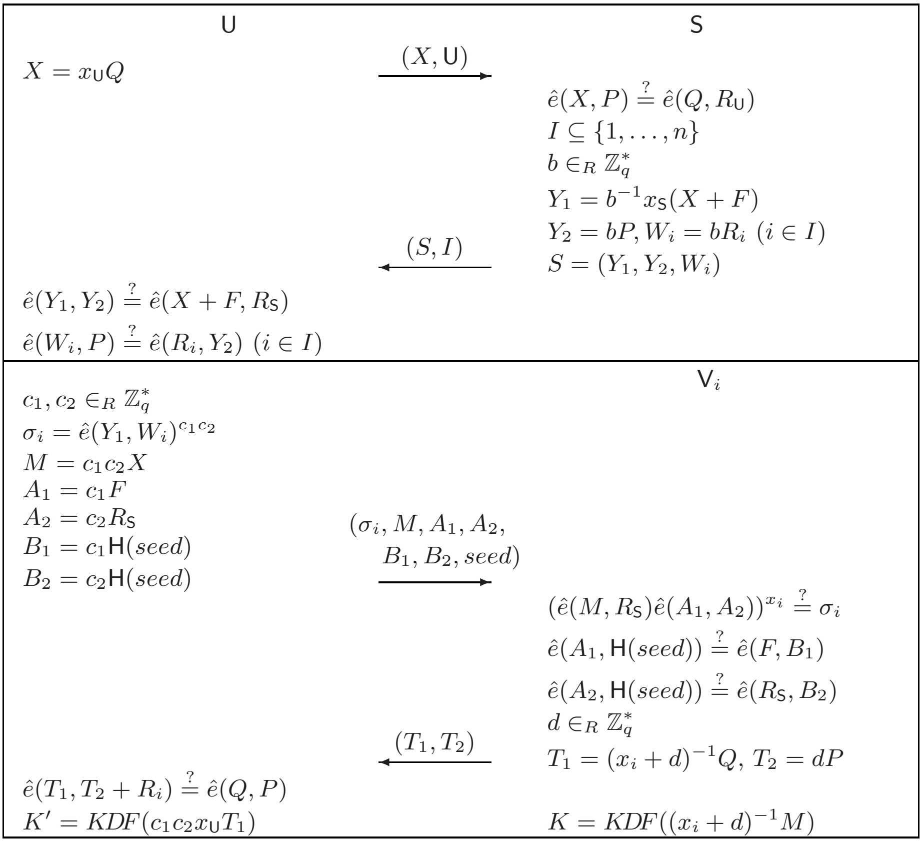
Key generation. The public/secret key pairs of S, U, and V_j are defined to be $(x_{\mathsf{S}}, R_{\mathsf{S}})$, $(x_{\mathsf{U}}, R_{\mathsf{U}})$, and (x_j, R_j) $(j = 1, \ldots, n)$, respectively, where $R_{\mathsf{S}} = x_{\mathsf{S}}P, R_{\mathsf{U}} = x_{\mathsf{U}}P, R_j = x_jP$. The secret keys x_{S}, x_{U}, and x_j $(j = 1, \ldots, n)$ are selected randomly from $\mathbb{Z}_q^*$, and the public keys are elements of group $\mathbb{G}_1$.

2.5 Basic Protocol

Verifiers in the basic protocol are designated by S. S knows all the verifier V_i's that U can access. In addition, the data have no time restrictions. They can be used as many times as wishes. Because this paper reports work in progress, we will only discuss the basic protocol. There are other schemes where U can specify verifier V_i's and schemes with time restrictions, i.e., the credential becomes invalid after a certain period. We should note that authentication between U and S is done by some other means not provided by the proposed scheme.

Request credential: Assume that user U would like to access V_i $(i \in I$, where $I \subseteq \{1, \ldots, n\})$. User U computes $X = x_{\mathsf{U}}Q$ and sends it together with U's identity as a request to S.

Receiving the request, S checks whether the secret information, x_{U}, is included in the request, $\hat{e}(X, P) \stackrel{?}{=} \hat{e}(Q, R_{\mathsf{U}})$. If no attempt at fraud is found, then S proceeds to issue a credential. It chooses b uniformly and randomly from $\mathbb{Z}_q^*$, and computes $Y_1 = b^{-1}x_{\mathsf{S}}(X + F)$ and $Y_2 = bP$. Then, S designates verifier list $I \subseteq \{1, \ldots, n\}$, and computes $W_i = bR_i$ $(i \in I)$. Finally, S sets $S = (Y_1, Y_2, W_i)$ and sends the credential (S, I) to U.



U		S
$X = x_{\mathsf{U}}Q$	$\xrightarrow{(X, \mathsf{U})}$	
		$\hat{e}(X, P) \overset{?}{=} \hat{e}(Q, R_{\mathsf{U}})$ $I \subseteq \{1, \ldots, n\}$ $b \in_R \mathbb{Z}_q^*$ $Y_1 = b^{-1}x_{\mathsf{S}}(X + F)$ $Y_2 = bP, W_i = bR_i \ (i \in I)$
	$\xleftarrow{(S, I)}$	$S = (Y_1, Y_2, W_i)$
$\hat{e}(Y_1, Y_2) \overset{?}{=} \hat{e}(X + F, R_{\mathsf{S}})$ $\hat{e}(W_i, P) \overset{?}{=} \hat{e}(R_i, Y_2) \ (i \in I)$		
		V_i
$c_1, c_2 \in_R \mathbb{Z}_q^*$ $\sigma_i = \hat{e}(Y_1, W_i)^{c_1 c_2}$ $M = c_1 c_2 X$ $A_1 = c_1 F$ $A_2 = c_2 R_{\mathsf{S}}$ $B_1 = c_1 \mathsf{H}(seed)$ $B_2 = c_2 \mathsf{H}(seed)$	$\xrightarrow{(\sigma_i, M, A_1, A_2, B_1, B_2, seed)}$	
		$(\hat{e}(M, R_{\mathsf{S}})\hat{e}(A_1, A_2))^{x_i} \overset{?}{=} \sigma_i$ $\hat{e}(A_1, \mathsf{H}(seed)) \overset{?}{=} \hat{e}(F, B_1)$ $\hat{e}(A_2, \mathsf{H}(seed)) \overset{?}{=} \hat{e}(R_{\mathsf{S}}, B_2)$ $d \in_R \mathbb{Z}_q^*$
	$\xleftarrow{(T_1, T_2)}$	$T_1 = (x_i + d)^{-1}Q,\ T_2 = dP$
$\hat{e}(T_1, T_2 + R_i) \overset{?}{=} \hat{e}(Q, P)$		
$K' = KDF(c_1 c_2 x_{\mathsf{U}} T_1)$		$K = KDF((x_i + d)^{-1}M)$

Fig. 1. Proposed protocol

User U verifies the received credential as $\hat{e}(Y_1, Y_2) \overset{?}{=} \hat{e}(X + F, R_{\mathsf{S}})$, $\hat{e}(W_i, P) \overset{?}{=} \hat{e}(R_i, Y_2)$.

Request for service: Assume that U would like to ask for service V_i, where i belongs to I. Then U chooses c_1, c_2 uniformly and randomly from $\mathbb{Z}_q^*$, and computes $\sigma_i = \hat{e}(Y_1, W_i)^{c_1 c_2}$, $M = c_1 c_2 X$, $A_1 = c_1 F$, $A_2 = c_2 R_{\mathsf{S}}$, $B_1 = c_1 \mathsf{H}(seed)$, and $B_2 = c_2 \mathsf{H}(seed)$. Whenever asking for a service, $\mathsf{H}(seed)$ makes a fresh generator from a random value to satisfy the non-transferability by U.

The validated ticket $(\sigma_i, M, A_1, A_2, B_1, B_2, seed)$ is sent to verifier V_i which checks whether the ticket has been correctly generated using a credential issued by S.

$$(\hat{e}(M, R_{\mathsf{S}})\hat{e}(A_1, A_2))^{x_i} \overset{?}{=} \sigma_i \tag{2.1}$$

$$\hat{e}(A_1, \mathsf{H}(seed)) \overset{?}{=} \hat{e}(F, B_1) \tag{2.2}$$

$$\hat{e}(A_2, \mathsf{H}(seed)) \overset{?}{=} \hat{e}(R_{\mathsf{S}}, B_2). \tag{2.3}$$

If no frauds are found, V_i then computes a session key.

Key exchange: V_i chooses d uniformly and randomly from $\mathbb{Z}_q^*$, and computes $T_1 = (x_i + d)^{-1}Q$, $T_2 = dP$, and $K = \mathit{KDF}((x_i + d)^{-1}M)$, where KDF is a key derivation function such as the KDF1 defined in IEEE Standard 1363-2000.

Then, data T_1 and T_2 are sent to U, and U checks the validity, $\hat{e}(T_1, T_2 + R_i) \overset{?}{=} \hat{e}(Q, P))$. If no frauds are found, U then computes session key $K' = \mathit{KDF}(c_1c_2x_{\mathsf{U}}T_1)$. If the protocol is carried out correctly, we have $K = K'$ and this will be used as the secret key between U and V_i. The flow for the protocol is shown in Fig. 1.

3 Security

Unforgeability. Forging a valid ticket may be attempted by many entities other than the targeted verifier (say V_1). Here, we will discuss unforgeability of the scheme by a legitimate user (say U).

If V_1 accepts the data $(\sigma_1, M, A_1, A_2, B_1, B_2, \mathit{seed})$ and U can generate the same session key with V_1, the forging attack succeeds. If there are no polynomial time algorithms forging a valid ticket succeeding with non-negligible probability, then the scheme is secure.

Clearly, under the CDH assumption, to obtain session key $K' = K = (x_1 + d)^{-1}M$ using the given $T_1 = (x_1 + d)^{-1}Q$, data M must have the form $M = yQ$ for some known $y \in \mathbb{Z}_q^*$. However, note that to pass verification equations (2.2) and (2.3), data (A_1, A_2) must have the form $A_1 = c_1F$ and $A_2 = c_2R_S$, for some known $c_1, c_2 \in \mathbb{Z}_q^*$. Now substitute $M = yQ$, $A_1 = c_1F$, $A_2 = c_2R_S$ into verification equation (2.1), then the left-hand side is $(\hat{e}(M, R_{\mathsf{S}})\hat{e}(A_1, A_2))^{x_i} = (\hat{e}(yQ + c_1c_2F, R_s))^{x_i}$. Note that $yQ + c_1c_2F = zP$ for some $z \in \mathbb{Z}_q^*$ because $yQ + c_1c_2F$ is an element of $\mathbb{G}_1$. However, finding parameter z is a discrete logarithm problem to U. Accordingly, to forge a valid σ_1, U has to solve the BDH problem, i.e., given $\langle P, zP, R_1 = x_1P, R_{\mathsf{S}} = x_{\mathsf{S}}P\rangle$, to compute $\hat{e}(P, P)^{zx_{\mathsf{S}}x_1}$. According to the complexity assumption, there are no polynomial time algorithms to solve the BDH problem. Therefore our scheme satisfies the Unforgeability.

Limitability. User U may request and obtain a credential from S. Then, U may want to forge the valid credential issued by S. Here, we are assuming that V_1 has not been designated in the credential issued by S.

U has the information, $R_{\mathsf{S}}, R_j(j \in \{1, \ldots, n\})$; x_{U} ; $Y_1, Y_2, W_i(i \in I, i \neq 1)$, apart from public parameters $P, Q, F \in \mathbb{G}_1$. Then, U would like to forge $(\sigma_1, M, A_1, A_2, B_1, B_2, \mathit{seed})$. In detail, using the data $\langle Y_1, W_i(i \in I, i \neq 1)\rangle$ issued by S, U can compute $\sigma_i^0 = \hat{e}(Y_1, W_i) = \hat{e}(x_{\mathsf{U}}Q, R_{\mathsf{S}})^{x_i}\hat{e}(F, R_{\mathsf{S}})^{x_i}$.

Both $\hat{e}(x_{\mathsf{U}}Q, R_{\mathsf{S}})^{x_i}$ and $\hat{e}(F, R_{\mathsf{S}})^{x_i}$ are BDH problems for user U. Therefore, σ_i^0 cannot be split. In fact, $\sigma_i^0 = \hat{e}(x_{\mathsf{U}}Q + F, R_{\mathsf{S}})^{x_i} = (\hat{e}(P, P)^{wx_{\mathsf{S}}})^{x_i}$ for some unknown $w \in \mathbb{Z}_q^*$. Let $g = \hat{e}(P, P)^{wx_{\mathsf{S}}} = \hat{e}(X + F, R_{\mathsf{S}}) \in \mathbb{G}_2$; this then implies that U has found g^{x_1} using the given g^{x_i} where $i \in I, i \neq 1$, which is a discrete logarithm problem in $\mathbb{G}_2$. Our scheme satisfies limitability according to the complexity assumption described in Section 2.3.

Non-transferability. The proof has been omitted due to limited space. The main point (e.g., case 2) is that, to successfully transfer the credential to another party (say, Tom) the transferred data must have passed verification and Tom can generate the same session key with V_i. According to the analysis of unforgeability, as Tom must know the value of $x_{\mathsf{U}}c_1c_2$, where he has selected c_1 and c_2 himself, U has to reveal his/her secret key x_{U} to Tom.

Unlinkability. Unlinkability will be proved under the decisional bilinear Diffie-Hellman (DBDH) assumption [5] in the full version of this paper. Note that unlinkability implies anonymity. That is to say, our scheme also satisfies anonymity.

4 Discussion

The contribution of our anonymous authenticated key exchange scheme can be summarized as follows. The basic protocol achieves **anonymous authenticated key exchange**, i.e., S issues the credential for U to share the key with V anonymously, i.e., U can share a secret key with V anonymously and securely; U does not necessarily leak any of her/his identity information to V but V can authenticate U.

We did not discuss additional properties of **hidden verifiers** or **time restrictions** and these properties will be fully discussed in the full version of the paper. The first implies that the proposed protocol can convert an open verifier to a hidden verifier for S; when U requests a credential of V, U chooses some services, and sends these to S with a random value to hide service. The second implies that the proposed protocol can attach time restrictions function easily; it only need changes generator F of the protocol into $\mathsf{H}(t)$, where t is the time information.

References

1. Boneh, D., Franklin, M.: Identity-based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
2. Camenisch, J., Lysyanskaya, A.: An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001)
3. Joux, A.: The Weil and Tate Pairings as Building Blocks for Public Key Cryptosystems. In: Fieker, C., Kohel, D.R. (eds.) Algorithmic Number Theory. LNCS, vol. 2369, pp. 20–32. Springer, Heidelberg (2002)
4. Ng, C.Y., Susilo, W., Mu, Y.: Designated Group Credentials. In: Proc. of ASIACCS 2006, pp. 59–65 (2006)
5. Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R.J.F. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)