International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:11, 2011

Account Management Method with Blind Signature Scheme

Ryu Watanabe and Yutaka Miyake

Abstract—Reducing the risk of information leaks is one of the most important functions of identity management systems. To achieve this purpose, Dey et al. have already proposed an account management method for a federated login system using a blind signature scheme. In order to ensure account anonymity for the authentication provider, referred to as an IDP (identity provider), a blind signature scheme is utilized to generate an authentication token on an authentication service and the token is sent to an IDP. However, there is a problem with the proposed system. Malicious users can establish multiple accounts on an IDP by requesting such accounts. As a measure to solve this problem, in this paper, the authors propose an account checking method that is performed before account generation.

Keywords—identity management, blind signature, privacy protection

I. Introduction

RECENT innovation in information technology (IT) and progress in the infrastructure for high-speed communication has led to the development of more flexible Internet services and various WEB services can now be provided to users such as on-line shopping, VoD services and social network services (SNS). In this situation, in order to enhance both the security and usability of WEB services, identity management technology for single sign-on (SSO) techniques is being utilized more than ever. In addition, many specifications and implementations are provided[1][2]. By using an SSO technique, only one authentication on an identity provider (IDP) can substitute authentication on each service provider (SP).

Once an IDP authenticates a user, it retains the user authentication status. When the user wants to use a service, the service provider delegates user authentication to the IDP. Then the IDP can advise the service of the user's authentication status. The SP receives the notification from the IDP and checks it, completing user authentication on the SP. If the user wants to use different services, each SP authenticates the user in the same manner. Therefore, users do not have to input each ID/PW pair and are freed from the nuisance of having to manage numerous ID/PW.

In this SSO technique, only IDPs know the user identity for authentication, therefore, they have to strictly manage and maintain such important information as they would for a user's account. One of the issues of this form of account management is a privacy problem. Usually, in order to avoid a linkability problem, the accounts on an IDP and a SP are federated using a pseudonym. However, if an IDP is cracked, the relationship between the accounts on the IDP and the pseudonym is revealed. Then, from this relationship, the user identity managed in an IDP can be linked to user activities on each service.

Ryu Watanabe and Yutaka Miyake belong to the KDDI R&D Laboratories., Inc. 356-8502 Kamifukuoka 2-1-15, Fujimino, Saitama, Japan.

As a measure to deal with this account problem, Dey et al. proposed an account management method called PseudoID. PseudoID is a federated login system that protects users from disclosure of private login data held by identity providers. Even though an IDP is cracked, the user identity for authentication cannot be revealed, thereby protecting user privacy. The PseudoID separates the functions on the IDP for this purpose. In PseudoID, an authority, which is dedicated to preserving user identity is prepared. This authority confirms users' identities and authenticates users. The IDP utilizes the results from the authority and then generates a user account for the user. In order to realize this function, the PseudoID utilizes a blind signature scheme. The authority signs an authentication token submitted from a user using a blind signature scheme, which means that the authority cannot obtain information about a user account on an IDP. As a result, in the event the authority is cracked, the user identity maintained on the authority and activities on SPs are not linked. Due to the fact that the PseudoID uses a blind signature scheme[4], the authority is called a blind signature service (BSS).

However, Pseudo ID still has a problem in that the authority does not check if a user has already established an account on the IDP. Therefore, a malicious user can establish multiple accounts on an IDP by submitting multiple requests to the BSS. One user having multiple accounts can cause a problem on some services.

In order to avoid this multiple accounts problem on PseudoID, the authors propose token management on the BSS. In our proposal, the BSS checks the status of a blinded token before signing a token. We also configure the sequence for account deletion on an IDP. When a user deletes his or her account on an IDP, the status of the blinded token is retained by the BSS. Therefore, the user can make new accounts on the IDP.

II. Related work

A. Blind signature

First, we describe the blind signature scheme, which is a key technique for the PseudoID. The blind signature scheme is an extension of the digital signature scheme as one application of public key cryptosystems. The digital signature scheme consists of the signing algorithm $S()$ with a private key, which belongs to the signer, and the verification algorithm

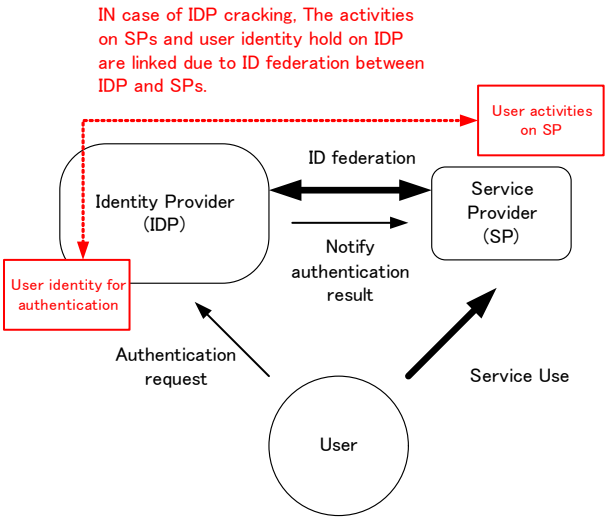

Fig. 1. Privacy problem on SSO by ID federation

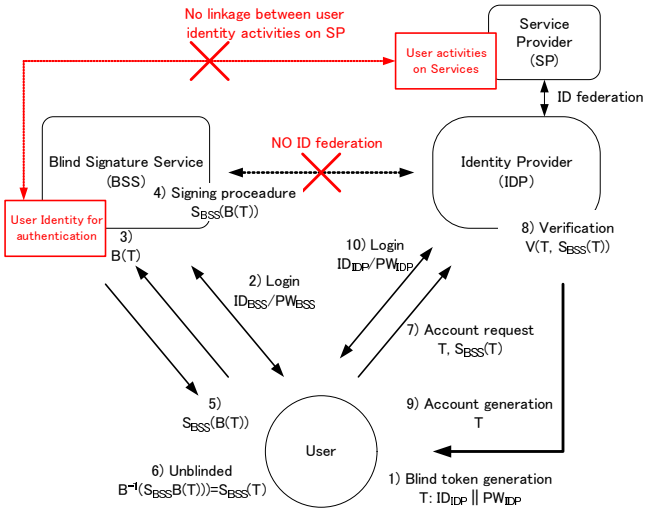

Fig. 2. Schematic representation of PseudoID account management

$V()$ with a public key, which is open to the public and paired with the private key. The signer appends his / her signature to a message with his / her private key, which only he / she knows. The verifier verifies the signed message with the public key. In the verification algorithm, with the message M, which is the signing target, and the signed message $S(M)$, the $V(M, S(M))$ is calculated for verification.

The blind signature scheme applies a blind messaging process to an ordinary signature scheme. The blinding function $B()$ is introduced and used for making messages unreadable. The blinding function $B()$ and the signature function $S()$ satisfies the equation given below. In addition, the $B^{-1}()$ is a reverse function of $B()$.

$$B^{-1}(S(B(m))) = B^{-1}(B(S(m))) = S(m) \qquad (1)$$

In a blind signature scheme, users want a signer to generate a signature without revealing his / her message to the signer. Therefore, the user encrypts the message using a blinding function and sends it to a signer. Then the signer generates a signature on the blinded message with its private key and returns it to the user. The user can unblind the signed message by using the relationship between the blinding function and signing function denoted in equation (1) and obtain the signature against the message. Then the user sends the signature to a verifier. The verifier can verify the signed message with the signer's public key.

B. PseudoID

In the current single sign-on scheme, an IDP manages user identity for account generation and the IDs on both the IDP and SPs are linked via an ID federation technique. Therefore, there is a security concern about user privacy. For example, if an IDP is cracked by malicious party, the user's identity held on the IDP and the activities on services are revealed through this linking between IDs. A conceptual representation of this problem is shown in Fig. 1.

TABLE I
DEFINITION OF SYMBOLS

Symbol	Definition
ID_X	User ID on entitiy X
PW_X	User password on entity X
$B()$	Blind function
$S_X(M)$	Signature for message M by entity X
$V(M, S_X(M))$	Verification function for Signature
R_X	Random number generated by entity X
$\|\|$	concatenation

To resolve this privacy problem, Dey et al. proposed an account management method referred to as "PseudoID". In the PseudoID method, the user identification function is delegated to a dedicated party called a blind signature service from an IDP. Between this dedicated party and the IDP, the IDs of both parties are not linked. Therefore, if the IDP or BSS is cracked, the risk of an identity leak is reduced. In order to perform this function, the PseudoID applies the blind signature scheme. Before explaining the PseudoID scheme in more detail, the scheme's assumptions are summarized. The BSS can make user accounts for itself by identifying users' identities and retaining them for user management. The BSS opens its public key and other parties can confirm its legitimacy through some operation.

On PseudoID, first, the user prepares an ID/PW pair for an IDP and the pair is used as token between a BSS and the IDP. Then the user blinds the token and sends it to the BSS. The BSS authenticates the use and then generates a signature against the token. The token is then returned to the user. The user unblinds the token and submits the signed token to the IDP. If the IDP can verify the signature for the token, the IDP generates a user account and accompanying password for the user based on the contents of the token. The concept and the sequence for ID generation are shown in Fig. 2. and Fig. 3, respectively. The definition of symbols in Fig. 2 and Fig. 3 is summarized in table 1.

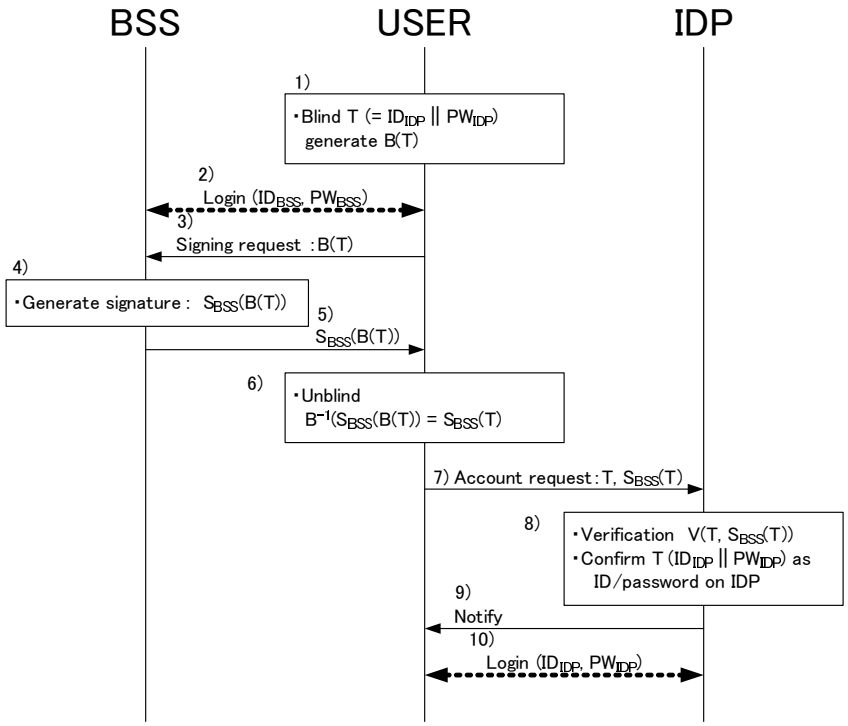

Fig. 3. Sequence of PseuoID account management

C. Problem with PseudoID

In the PseudoID scheme, in order to realize user privacy protection, the functions for user account generation are divided between the BSS and IDP. Therefore, the anonymity of user accounts on IDPs is maintained. However, this feature causes a different problem in that users can establish multiple accounts on a single IDP. In the PseudoID scheme, the ID/PW pair on an IDP is hidden from the BSS using a blinding function. Therefore, if a user requests the BSS to sign the tokens, which each have a different ID/PW pair, the user can generate multiple accounts on the targeted IDP (Fig. 4). This action is undesirable from the IDP's viewpoint.

III. Our Proposal

As a measure against the problem described in the previous section, the authors propose a modification to the PseudoID method. The concept of our proposal is simple. In order to avoid multiple account generation on an IDP, token generation management is also introduced on a BSS. If a user requests multiple token generation (signing to multiple tokens) to a BSS, the BSS then checks the status of token release to the user. If the BSS has already issued a token to the user, the BSS refuses to sign the user's token.

A. Requirements

For our proposal, we took usability and security into account. The requirements are shown below.

1) The accounts on BSS and IDP must have no relationship.

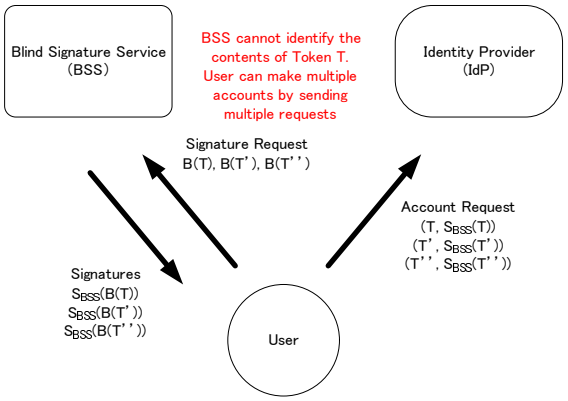

Fig. 4. Problem of multiple account generation on PseudoID

2) Users can regenerate their accounts on the IDP.

The first requirement is given for security reasons and the second one for operational usability. Identical to the original PseudoID scheme, in order to realize user privacy protection, the blind signature technique is used for the proposed system's procedures. In addition, in the proposed scheme, the BSS cannot sign a second token when requested to do so by the same user. Therefore, in order to regenerate an account on the IDP, the modified scheme also has a delete function.

In addition, in our proposal, it is assumed that the BSS and IDP have established a trust relationship and trust each other. The public keys of both parties are known to the public and

they can confirm their legitimacy easily. For communication between users and the BSS or IDP, a secure connection can be utilized and messages are not revealed to a third party.

B. Account generation

In the original PseudoID method, an ID/PW pair is used for the content of the token, which will be signed by the BSS. Instead of the ID/PW pair, a unique random number, which is generated by the IDP, is utilized in our scheme. This random number is used as an identifier on the IDP for the account generation procedure. The procedure is described in detail below and the procedure sequence is shown in Fig. 5.

1) User requests the IDP to generate user accounts on the IDP.
2) The IDP generates a unique random number R_{IDP} and then sends it to the user. This number and its generation time are recorded on the IDP. The number has a lifetime. If the time exceeds the specified lifetime, the number expires.
3) The user generates [and sends] a request token ($T = R_{IDP}$) to the BSS and blinds it with the blinding function ($B(T)$).
4) The user logs in to the BSS with his / her ID/PW pair.
5) The user requests the BSS to sign the token $B(T)$.
6) The BSS checks the user's token issue status. If the status is "not issued", the BSS generates a signature for the token $S_{IDP}(B(T))$ and changes the token issue status. If the status is "issued", then the BSS aborts the procedure.
7) The BSS sends the signature to the user.
8) The user unblinds the token with the function $B^{-1}()$ and obtains the signature $S_{IDP}(T)$.
9) The user sends the token and its signature $(T, S_{IDP(T)})$ to the IDP.
10) The IDP obtains the R_{IDP} from T and checks the status of the random number R_{IDP}. If the status is not "expired" or "used", the IDP validates the signature. If the validity of the signature is confirmed, the status of the random number R_{IDP} is changed to "used".
11) Then the IDP notifies the user that permission has been obtained for account generation.
12) The user registers the ID/PW pair on the IDP and uses it for logging in later.

C. Account deletion

The blind signature scheme is also used for account deletion. In the deletion procedure, the BSS generates a unique random number for identification. In order to avoid reuse of the token by a malicious user, the number is managed on the BSS. The deletion procedure is described below and the deletion sequence is shown in Fig. 6.

1) The user logs in to the BSS.
2) The user notifies that he/she wishes to delete the IDP account that had been set up previously.
3) The BSS confirms that the token status is "issued" and finds the $B(T)$ using the user ID on the BSS (ID_{BSS})

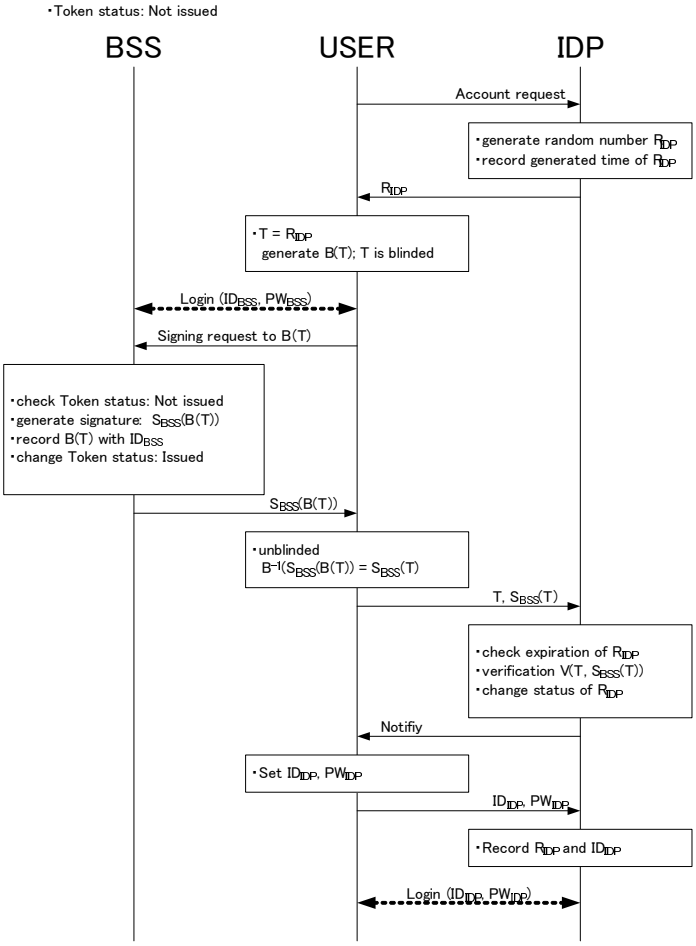

Fig. 5. Account generation sequence of the proposed method

that was reported by the user at the time when the account was generated. Then the BSS generates a unique random number (R_{BSS}). The number is recorded with the user ID (ID_{BSS}) and $B(T)$.
4) The BSS sends the $B(T)$ and R_{BSS} to the user.
5) The user generates a deletion token $D = R_{BSS}$ and blinds the D as $B(D)$.
6) The user logs in to the IDP.
7) The user notifies the deletion of the account on the IDP and submits the blinded token ($B(D)$) to the IDP.
8) The IDP signs the blinded token ($B(D)$). By using the user ID (ID_{IDP}), the IDP searches for the R_{IDP} and change the status of the unique random number to "deleted". Finally, the IDP deletes the user's account.
9) The IDP submits the signed token ($S_{IDP}(B(D))$) to the user.
10) The user unblinds the token and obtains the signature ($S_{IDP}(D)$).
11) The user submits the BSS $(D, S_{IDP}(D))$.
12) The BSS checks the D and validates the signature. If legitimacy is confirmed, the BSS changes the token issue status for the user to "not issued".

By performing this procedure, account deletion can be carried out without revealing the account information to the BSS. This means that users can regenerate their accounts on the IDP and their privacy is protected.

IV. Discussion

In the proposed scheme, a secure account deletion procedure is introduced for user account management on the IDP and the BSS executes token management for the procedure. Therefore, the second requirement described in the third section is satisfied. Moreover, by using the blind signature method for both account generation and deletion on the IDP, the the risk of the user's identity being leaked is reduced because the IDs on the BSS and IDP are not linked. Therefore, as a result, user privacy is protected and the first requirement is also satisfied. By use of the proposed method, multiple account generation can be prevented and it is probable that account management costs can also be reduced.

The random numbers, which are generated on the BSS and IDP, are used to prevent the reuse of blinded tokens. If these numbers are not used, users can reuse the tokens signed by the BSS or IDP and establish multiple accounts on the IDP without permission from the BSS. The random numbers are unique to the BSS or IDP. Once the numbers are used, the same numbers are cancelled and cannot be used in the future. Therefore, a token with an expired number cannot be reused. In addition, the fact that the numbers have a lifetime also contributes to secure operation with our procedure because it is extremely difficult for malicious users to find active numbers.

The main purpose of our proposed scheme is to prevent users from generating multiple accounts. Thus, the procedures have been designed for this purpose. Therefore, the requirement for user privacy to be protected is relaxed compared to the original PseudoID. In the original PseudoID, the basic function of the BSS is generating a signature for the tokens following a user request. Therefore, the BSS does not care which IDP users want to set up accounts in. In contrast, in our scheme, in order to recover the token condition on the BSS, the BSS knows which IDP the users have set up their accounts in.

V. Conclusion

In this paper, the authors proposed an extension of the account management method named PseudoID. We introduced token generation management on the BSS, which is used for account generation on the IDP as an additional function. The proposed method prevents multiple account generation on an IDP. In the account deletion procedure, a blind signature scheme is used and prevents any linkage or relationship between accounts on both the BSS and IDP. By using this method, therefore, the risk of identity leak is reduced. In future, based on our proposal, we will implement a prototype and conduct a performance test. Moreover, we will investigate an additional extension.

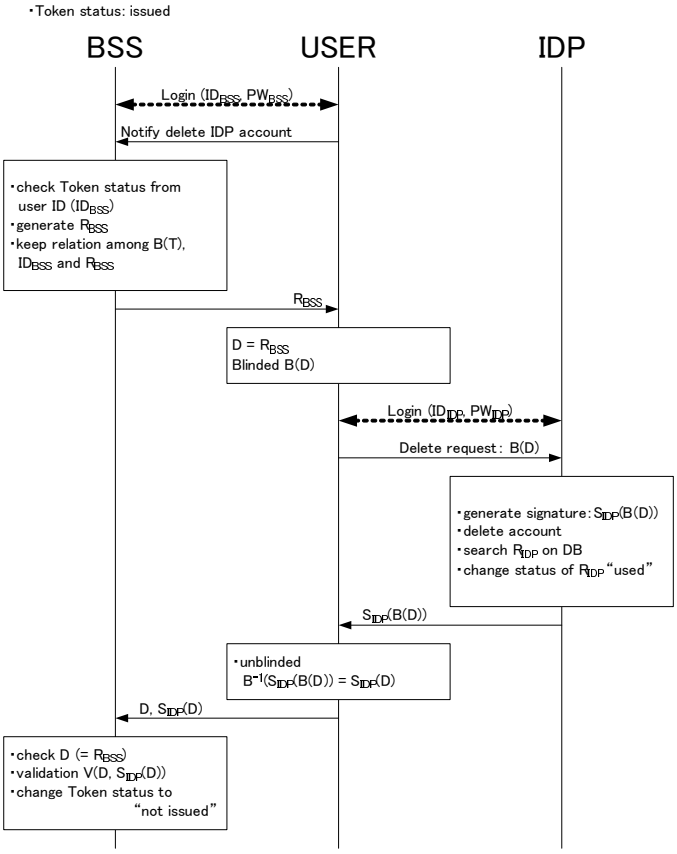

Fig. 6. Account deletion sequence of the proposed method

References

[1] Security Assertio Markup Language (SAML) V2.0, OASIS (2005), http://www.oasis-open.org/specs/index.php#samlv2.0
[2] OpenID Authentication 2.0 - Final, OpenID Foundation, (2007), http://openid.net/specs/openid-authentication-2_0.txt
[3] Arkajit Dey and Stephen Weis, "PseudoID: Enhancing Privacy in Federated Login," *Proc. 3rd Hot Topics in Privacy Enhancing Technologies(HotPETs 2010)*, pp.95-107 (2010).
[4] David Chaum, "Blind signatures for untraceable payments," *CRYPTO,* pp.199-203 (1982).
[5] Whitfield Diffie and Martin E. Hellman, "New directions in cryptography," *Trans. on Information Theory, IEEE*, Vol. 22, Issue 6, pp. 644-654 (1976).