LETTER

How to Preserve User Anonymity in Password-Based Anonymous Authentication Scheme

SeongHan SHIN[†a)], ***Nonmember and*** **Kazukuni KOBARA**[†], ***Member***

SUMMARY A purpose of password-based anonymous authentication schemes is to provide not only password-based authentication but also user anonymity. In [19], Yang et al., proposed a password-based anonymous authentication scheme (we call it YZWB10 scheme) using the password-protected credentials. In this paper, we discuss user anonymity of the YZWB10 scheme [19] against a third-party attacker, who is much weaker than a malicious server. First, we show that a third-party attacker in the YZWB10 scheme can specify which user actually sent the login request to the server. This attack also indicates that the attacker can link different login requests to be sent later by the same user. Second, we give an *effective* countermeasure to this attack which does not require any security for storing users' password-protected credentials.

key words: *password, authentication, user anonymity*

1. Introduction

A main purpose of password-based anonymous authentication schemes is to provide not only password-based authentication but also user anonymity. So far, several schemes (e.g., [15], [17]–[19]) have been proposed in different settings. Some potential applications of these schemes include whistle-blowing from insiders, questionnaire to qualified people, anonymous counseling, and so on.

In [18], Yang et al., proposed a new password-based anonymous authentication scheme using the password-protected credentials. This scheme is constructed on Camenisch's signature [6] for instantiating users' authentication credentials, and Paillier encryption [12] for server's homomorphic encryption. Some elements of the authentication credential (i.e., signature on user's identity) are encrypted with user's password, while other elements are encrypted with server's public-key (homomorphic) encryption. For better efficiency, Yang et al., [19] proposed another password-based anonymous authentication scheme (we call it YZWB10 scheme) which is based on the BBS+ signature [1] (instead of Camenisch's signature [6]) and the ElGamal encryption (instead of Paillier encryption [12]). The main idea of [18], [19] is to restrict the signature verifiability to server only via a zero-knowledge proof of knowledge protocol. As a distinguishing feature of [18], [19], Yang et al., claimed that the password-protected credentials must not require any secure storage facility for usability of the schemes. Recently, Shin et al., [16] showed that the YZWB10 scheme does not provide unlinkability against a malicious server.

Manuscript received August 18, 2017.
Manuscript revised November 14, 2017.
Manuscript publicized December 13, 2017.
[†]The authors are with the Information Technology Research Institute (ITRI), National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, 135–0064 Japan.
a) E-mail: seonghan.shin@aist.go.jp
DOI: 10.1587/transinf.2017EDL8183

1.1 Our Contributions

In Table 1, we summarize anonymity levels of password-based anonymous authentication schemes (including the YZWB10 scheme [19]) which have being standardized in ISO/IEC 20009-4 [9]. In this paper, we discuss user anonymity of the YZWB10 scheme [19] against a passive/active third-party attacker, who is much weaker than a malicious server. Our contributions are twofold (see also Table 1). First, we show that a third-party attacker in the YZWB10 scheme can specify which user actually sent the login request to the server (Sect. 4). This attack also indicates that the attacker can link different login requests to be sent later by the same user. From this attack, it is clear that the YZWB10 scheme (both the basic and extended schemes) [19] does not provide user anonymity against a third-party attacker. Second, we give an *effective* countermeasure to the attack of Sect. 4 which does not require any security for storing users' password-protected credentials (Sect. 5).

2. Preliminaries

2.1 Notations

First, $a \in_R S$ means that a is randomly chosen from S. Let G_1, G_2, G_T be cyclic groups of prime q. Let g be a generator of G_1, and h be a generator of G_2. A bilinear map $e : G_1 \times G_2 \rightarrow G_T$ has the following properties: a) Bilinear: $\forall u \in G_1, v \in G_2$ and $x, y \in_R Z_q$, $e(u^x, v^y) = e(u, v)^{xy}$, and b) Non-degenerate: $e(g, h) \neq 1$.

2.2 BBS+ Signature

In [1], Au et al., modified the BBS group signature [3] for their dynamic k-times anonymous authentication scheme. The modified signature (called, BBS+ signature) is a signature scheme with efficient protocols for issuing a signature on a committed value, and for proving zero-knowledge of a signature on a committed value.

A public key of the BBS+ signature scheme is $(W = h^{\chi}, h \in G_2, a, b, d \in G_1)$, and a private key is $(\chi \in Z_q)$.

Table 1 Anonymity levels of password-based anonymous authentication schemes (included in ISO/IEC 20009-4 [9]) where our contributions are shown in **bold**

Password-based anonymous authentication schemes	Anonymity against		
	passive/active third-party attacker	semi-honest server	malicious server
[17]	Yes	Yes	Yes [1]
[15]	Yes	Yes	No
YZWB10 [19]	**No (Sect. 4)** ⇝ **Yes (Sect. 5)**	Yes	No [2]

[1]: There is a tradeoff between anonymity and client-side authentication
[2]: Due to [16]

A BBS+ signature signed on a message m is defined by (M, k, s) where $k, s \in_R Z_q$ and $M = (a^m \cdot b^s \cdot d)^{1/(k+\chi)} \in G_1$.

The BBS+ signature (M, k, s, m) is verified with respect to the public key as $e(M, W \cdot h^k) = e(a, h)^m \cdot e(b, h)^s \cdot e(d, h)$. This verification can be carried out in a zero-knowledge proof of knowledge protocol for showing possession of a signature. For more details, see [1].

3. YZWB10 Scheme

3.1 R-BBS Signature

As a main building block for the YZWB10 scheme, Yang et al., [19] also proposed a R-BBS signature which is a randomized version of the BBS+ signature [1]. Hereafter, the R-BBS signature is denoted by Π_{R-BBS}.

Instead of (M, k, s) of the BBS+ signature, a prover has in possession of $(M, k, \gamma, e(B, h))$ where $M = (a^u \cdot b^s \cdot d)^{1/(k+\chi)}$, u is a user's identity, $r \in_R Z_q$, $\gamma = r^{-1} \bmod q$, and $B = b^{r \cdot s}$. Note that it holds

$$e(B, h) = \left(\frac{e(M, W \cdot h^k)}{e(a, h)^u \cdot e(d, h)}\right)^r . \tag{1}$$

Let $g_0, g_1 \in G_1$ be pre-defined parameters. First, the prover chooses $\alpha, r_u, r_k, r_\gamma, r_\alpha, r_{\tilde{\alpha}} \in_R Z_q$, and then computes $\mathsf{Cmt}(\Pi_{R-BBS}) = \{T_1, T_2, R_1, R_2, R_3\}$ as follows:

$$\begin{aligned} T_1 &= M \cdot g_0^\alpha, \quad T_2 = g_1^\alpha, \\ R_1 &= \left(\frac{1}{e(T_1, h)}\right)^{r_k} \cdot e(a, h)^{r_u} \cdot e(B, h)^{r_\gamma} \cdot e(g_0, W)^{r_\alpha} \\ &\quad \cdot e(g_0, h)^{r_{\tilde{\alpha}}}, \\ R_2 &= g_1^{r_\alpha}, \quad R_3 = \left(\frac{1}{T_2}\right)^{r_k} \cdot g_1^{r_{\tilde{\alpha}}} . \end{aligned}$$

The prover sends $\mathsf{Cmt}(\Pi_{R-BBS})$ to the verifier, who sends back a challenge $c \in_R Z_q$. Upon receipt of the challenge, the prover computes $\mathsf{Res}(\Pi_{R-BBS}) = \{s_u, s_\gamma, s_k, s_\alpha, s_{\tilde{\alpha}}\}$ as follows: $s_u = r_u + c \cdot u$, $s_\gamma = r_\gamma + c \cdot \gamma$, $s_k = r_k + c \cdot k$, $s_\alpha = r_\alpha + c \cdot \alpha$, $s_{\tilde{\alpha}} = r_{\tilde{\alpha}} + c \cdot \tilde{\alpha}$, where $\tilde{\alpha} = \alpha \cdot k$. The prover sends $\mathsf{Res}(\Pi_{R-BBS})$ to the verifier, who accepts if all of the followings hold

$$R_2 \cdot T_2^c = g_1^{s_\alpha}, \qquad R_3 = \left(\frac{1}{T_2}\right)^{s_k} \cdot g_1^{s_{\tilde{\alpha}}},$$

$$R_1 \cdot \left(\frac{e(T_1, W)}{e(d, h)}\right)^c = \left(\frac{1}{e(T_1, h)}\right)^{s_k} \cdot e(a, h)^{s_u} \cdot e(B, h)^{s_\gamma} \cdot e(g_0, W)^{s_\alpha} \cdot e(g_0, h)^{s_{\tilde{\alpha}}} .$$

According to [19], the above R-BBS signature is an honest-verifier zero-knowledge proof of knowledge of a tuple (M, k, γ, u) subject to $e(M, W \cdot h^k) = e(a, h)^u \cdot e(B, h)^\gamma \cdot e(d, h)$.

3.2 Basic Scheme

Here, we describe a basic scheme of the YZWB10 scheme [19]. The basic scheme consists of **Setup**, **Registration** and **Authentication Protocol**.

3.2.1 Setup

In order to set up the system parameters, the server does the followings: a) It sets up the public key for the BBS+ signature as $(W = h^\chi, h \in G_2, a, b, d \in G_1)$ and the private key as $(\chi \in Z_q)$; b) It publishes $g, g_0, g_1 \in G_1$ as a part of the public parameters; c) It selects a public/privake key pair for the ElGamal encryption, and its encryption and decryption are denoted by $\mathsf{E}(\cdot)$ and $\mathsf{D}(\cdot)$, respectively. The ElGamal encryption is used as a multiplicative homomorphic encryption scheme; d) It chooses a hash function $\mathsf{H} : \{0,1\}^* \rightarrow \{0,1\}^{\kappa_0}$ and a MAC $\mathsf{MAC} : \{0,1\}^{\kappa_1} \times G_1^2 \rightarrow \{0,1\}^{\kappa_1}$ where κ_0, κ_1 are appropriate numbers.

3.2.2 Registration

In the basic scheme, all users need to register to the server in advance, for each getting an authentication credential. The server issues each user u_i a credential, which is a BBS+ signature (M_i, k_i, s_i) signed on the user identity u_i. Upon receipt of the credential, the user protects (M_i, k_i) using a symmetric key encryption with a key, derived from his/her password pw_i, i.e., $[M_i, k_i]_{pw_i}$; and encrypts s_i using the server's public key, i.e., $\mathsf{E}(s_i)$. The password-protected credential is $C_i =< u_i, [M_i, k_i]_{pw_i}, \mathsf{E}(s_i) >$. Finally, the user puts the password-protected credential C_i to his/her preferred storage, e.g., handphone, USB flash memory, or public facilities/directories.

3.2.3 Authentication Protocol

Suppose that a user u_i has the password-protected credential $C_i =< u_i, [M_i, k_i]_{pw_i}, \mathsf{E}(s_i) >$ available at the point of login. Below is the authentication protocol between the user u_i and the server.

Step 1. The user u_i does the followings: 1) The user recovers (M_i, k_i) from $[M_i, k_i]_{pw_i}$ with his/her password pw_i; 2) The user chooses $r \in_R Z_q$ to randomize $\mathsf{E}(s_i)$ by computing $s^* = \mathsf{E}(r) \cdot \mathsf{E}(s_i)$; 3) The user chooses $x \in_R Z_q$ and computes $X = g^x$; 4) The user chooses $N_A \in_R \{0,1\}^{\kappa_1}$ and computes $N_A^* = \mathsf{E}(N_A)$; 5) The user computes $\mathsf{Cmt}(\Pi_{R-BBS})$ using the R-BBS signature over $(M_i, k_i, \gamma = r^{-1} \pmod q), u_i)$; Finally, the user sends $s^*, X, N_A^*, \mathsf{Cmt}(\Pi_{R-BBS})$ to the server as a login request.

Step 2. Upon receipt of the login request, the server does the followings: 1) The server computes $r \cdot s_i = \mathsf{D}(s^*)$, due to the multiplicative homomorphic property of ElGamal, and $B = b^{r \cdot s_i}$; 2) The server chooses $y \in_R Z_q$ and computes $Y = g^y$; 3) The server computes $N_A = \mathsf{D}(N_A^*)$ and $V = \mathsf{MAC}(N_A, Y, X)$; 4) The server chooses $N_B \in_R Z_q$, and sends back N_B, Y, V to the user.

Step 3. The user u_i does the followings: 1) The user validates V, and aborts if invalid; 2) By taking N_B as a challenge, the user computes and sends $\mathsf{Res}(\Pi_{R-BBS})$ to the server; 3) The user ends the protocol by computing a shared key $sk = \mathsf{H}(N_A, N_B, Y^x)$.

Step 4. The server computes $sk = \mathsf{H}(N_A, N_B, X^y)$ upon verification of $\mathsf{Res}(\Pi_{R-BBS})$.

Note that $\mathsf{Cmt}(\Pi_{R-BBS})$ in **Step 1** can be computed by the user u_i, who does not know $r \cdot s_i$, since the user computes $e(B, h)$ as Eq. (1). In the above, the user u_i authenticates to the server by showing the possession of a correct credential while authentication of the server depends on the ElGamal encryption. In [19], Yang et al., also extended the basic scheme to support membership withdrawal by using the dynamic accumulator [11] (as in [1]).

4. User Anonymity against Third-Party Attacker

In [19], Yang et al., claimed that the YZWB10 scheme provides unlinkability against a server, who is much more powerful than an outside attacker, in the sense that the server cannot link different logins made by the same user. In this section, we show that a third-party attacker can specify which user sent the login request. Actually, this is enough for the third-party attacker to link different login requests sent by the same user.

4.1 Linkability of Third-Party Attacker

For clarity, suppose that there are only two users u_1 and u_2 whose password-protected credentials ($C_1 =< u_1, [M_1, k_1]_{pw_1}, \mathsf{E}(s_1) >$ for user u_1 and $C_2 =< u_2, [M_2, k_2]_{pw_2}, \mathsf{E}(s_2) >$ for user u_2) are entrusted to a public directory. In [19], Yang et al., clearly claimed that the password-protected credentials must not require any secure facility for storage and they can be entrusted to any portable devices, even public directories.

First, the attacker chooses $t \in_R Z_q$, computes $\mathsf{E}(t)$, and then replaces $C_1 =< u_1, [M_1, k_1]_{pw_1}, \mathsf{E}(s_1) >$ with $C_1' =< u_1, [M_1, k_1]_{pw_1}, \mathsf{E}(s_1) \cdot \mathsf{E}(t) >$. Below is the authentication protocol between the server and the user u_1, who has $C_1' =< u_1, [M_1, k_1]_{pw_1}, \mathsf{E}(s_1) \cdot \mathsf{E}(t) >$. In the authentication protocol, the third-party attacker just eavesdrops the communications between the user u_1 and the server. Of course, the attacker does not know which user is about to perform the protocol at the starting point of this protocol.

Step 1'. The user u_1 does the followings: 1) The user u_1 recovers (M_1, k_1) from $[M_1, k_1]_{pw_1}$ with his/her password pw_1; 2) The user u_1 chooses $r \in_R Z_q$ to randomize $\mathsf{E}(s_1) \cdot \mathsf{E}(t)$ by computing $s^* = \mathsf{E}(r) \cdot \mathsf{E}(s_1) \cdot \mathsf{E}(t)$; 3)–4) These are same as in **Step 1** of Sect. 3.2.3; 5) The user u_1 computes $\mathsf{Cmt}(\Pi_{R-BBS}) = \{T_1, T_2, R_1, R_2, R_3\}$ using the R-BBS signature over $(M_1, k_1, \gamma = r^{-1}, u_1)$ as follows:

$$T_1 = M_1 \cdot g_0^{\alpha}, T_2 = g_1^{\alpha}, R_2 = g_1^{r_\alpha}, R_3 = \left(\frac{1}{T_2}\right)^{r_k} \cdot g_1^{r_{\tilde{\alpha}}}$$

$$R_1 = \left(\frac{1}{e(T_1, h)}\right)^{r_k} \cdot e(a, h)^{r_u} \cdot e(B, h)^{r_\gamma} \cdot e(g_0, W)^{r_\alpha} \cdot e(g_0, h)^{r_{\tilde{\alpha}}}, \text{ where}$$

$$e(B, h) = \left(\frac{e(M_1, W \cdot h^{k_1})}{e(a, h)^{u_1} \cdot e(d, h)}\right)^r ;$$

Finally, the user u_1 sends s^*, X, N_A^*, $\mathsf{Cmt}(\Pi_{R-BBS})$ to the server as a login request.

Step 2'. Upon receipt of the login request, the server does the followings: 1) The server computes $r \cdot s_1 \cdot t = \mathsf{D}(s^*)$, due to the multiplicative homomorphic property of ElGamal, and $B' = b^{r \cdot s_1 \cdot t}$; 2)–4) These are same as in **Step 2** of Sect. 3.2.3.

Step 3'. The user u_1 does the followings: 1) This is same as in **Step 3** of Sect. 3.2.3; 2) By taking N_B as a challenge (i.e., $c = N_B$), the user u_1 computes $\mathsf{Res}(\Pi_{R-BBS}) = \{s_u, s_\gamma, s_k, s_\alpha, s_{\tilde{\alpha}}\}$ as follows: $s_u = r_u + c \cdot u_1$, $s_\gamma = r_\gamma + c \cdot \gamma$, $s_k = r_k + c \cdot k_1$, $s_\alpha = r_\alpha + c \cdot \alpha$, $s_{\tilde{\alpha}} = r_{\tilde{\alpha}} + c \cdot \tilde{\alpha}$, where $\tilde{\alpha} = \alpha \cdot k_1$, and then sends $\mathsf{Res}(\Pi_{R-BBS})$ to the server; 3) This is same as in **Step 3** of Sect. 3.2.3.

Step 4'. This is same as in **Step 4** of Sect. 3.2.3.

If the server aborts the protocol (i.e., $\mathsf{Res}(\Pi_{R-BBS})$ is invalid) in **Step 4'**, the attacker gets to know that the user who has just sent the login request is user u_1. Otherwise, the attacker comes to a conclusion that the user who has just sent the login request is user u_2. The invalidity of $\mathsf{Res}(\Pi_{R-BBS})$ in **Step 4'** can be easily checked from the following inequality:

$$R_1 \cdot \left(\frac{e(T_1, W)}{e(d, h)}\right)^c \neq \left(\frac{1}{e(T_1, h)}\right)^{s_k} \cdot e(a, h)^{s_u} \cdot e(B', h)^{s_\gamma} \cdot e(g_0, W)^{s_\alpha} \cdot e(g_0, h)^{s_{\tilde{\alpha}}} .$$

Remark 4.1: Suppose that an attacker replaces $[M_1, k_1]_{pw_1}$ with a random value t. After decrypting t with his/her password pw_1, the user u_1 gets a pair (M_1', k_1'). If the user can check formats of $M_1' \notin G_1$ or $k_1' \notin Z_q$, user u_1 can notice that t is not a correct one and abort the authentication protocol. In that case, the attacker can not break user anonymity. In order to keep formats of every elements consistent in the above attack, the attacker only randomizes the ciphertext of s_1 to $\mathsf{E}(s_1) \cdot \mathsf{E}(t)$ by using the homomorphic property of $\mathsf{E}(\cdot)$. This attack succeeds with probability 1, and the user u_1 cannot notice the change of $\mathsf{E}(s_1)$ since the password-protected credentials must not require any secure facility for storage (as claimed in [19]), and both $\mathsf{E}(s_1)$ and $\mathsf{E}(s_1) \cdot \mathsf{E}(t)$ have the same format.

4.2 Discussions

In the attack of Sect. 4.1, the third-party attacker can specify the user u_1 and u_2 with probability 1 by just eavesdropping the communications between the user and the server after replacing the password-protected credential C_1 with C'_1. This attack indicates that the attacker can link different login requests to be sent later by the user u_1. The main reason why the attack of Sect. 4.1 is possible is that the user can not check the integrity of $\mathsf{E}(s_1)$, at the same time, the server can not recover s_1 from the randomized s^*. The attack of Sect. 4.1 can be used to the YZWB10 scheme for many users u_i ($i > 2$). For example, if $|u_i| = 8$ and a user performs the **Authentication Protocol** of Sect. 3.2.3 consecutively, a third-party attacker can specify the user with probability 1 after repeating 3 times the attack of Sect. 4.1.

Also, the attack of Sect. 4.1 can be directly applied to the extended scheme to support membership withdrawal (i.e., Sect. 4.3 of [19]) because it is just addition of the basic scheme and Nguyen's dynamic accumulator [11]. In the extended scheme, the password-protected credential is the form of $C_i = < u_i, [M_i]_{pw_i}, k_i, w_i, \mathsf{E}(s_i) >$ where k_i is not encrypted with the password and is used to publish the accumulator Λ, and w_i is a witness of k_i for the dynamic accumulator [11]. One can see that this change is completely irrelevant to the attack of Sect. 4.1.

From the above, it is clear that the YZWB10 scheme (both the basic and extended schemes) [19] does *not* provide user anonymity against a third-party attacker.

5. A Countermeasure

A naive countermeasure to the attack of Sect. 4 is to use integrity-preserving portable devices or public directories for storing users' password-protected credentials. However, it is contrary to a distinguishing feature of the YZWB10 scheme [19] that the password-protected credentials must not require any secure facility for storage (on the user side).

In this section, we give a simple and *effective* countermeasure to the attack of Sect. 4 (i.e., another basic scheme to be described below) which does not require any security for storing users' password-protected credentials.

5.1 Basic Scheme

5.1.1 Setup

This is same as in **Setup** of Sect. 3.2.1.

5.1.2 Registration

This is same as in **Registration** of Sect. 3.2.2. In addition, the server stores the password-protected credentials $\{C_i\}_i$ for all users u_i locally.

5.1.3 Authentication Protocol

Suppose that a user u_i has the password-protected credential $C_i = < u_i, [M_i, k_i]_{pw_i}, \mathsf{E}(s_i) >$ available at the point of login. Below is the authentication protocol between the user u_i and the server.

Step 1. This is same as in **Step 1** of Sect. 3.2.3.
Step 2. Upon receipt of the login request, the server does the followings: 1)–2) These are same as in **Step 2** of Sect. 3.2.3; 3) The server computes $N_A = \mathsf{D}(N_A^*)$ and $V = \mathsf{MAC}(N_A, Y, X, \{C_i\}_i)$ where $\{C_i\}_i$ are the password-protected credentials (for all users) stored locally; 4) This is same as in **Step 2** of Sect. 3.2.3.
Step 3. This is same as in **Step 3** of Sect. 3.2.3.
Step 4. This is same as in **Step 4** of Sect. 3.2.3.

Remark 5.1: By using 'Secure CBC-MAC for arbitrary-length messages' (Chapter 4.4 of [10]) or HMAC ([4], [7], Chapter 5.3.2 of [10]) for MAC, one can obtain a fixed-length tag V whose size is independent of the message size. Another candidates for such MAC include UMAC [5], [13], OMAC [8] and Poly1305 [2], [14].

Remark 5.2: Though the above countermeasure is effective to the attack of Sect. 4, it is not practical when the number of users is large. This is the reason why each user must download all the password-protected credentials $\{C_i\}_i$, if they are entrusted to a public directory, or must store all $\{C_i\}_i$ locally, if they are entrusted to a user's portable device, in order to validate V. Actually, making the YZWB10 scheme [19] secure and efficient (i.e., independent of the number of users) would be a challenging task.

5.2 Discussions

In the basic scheme of Sect. 5.1, the user u_i can check the integrity of $\{C_i\}_i$ (including $\mathsf{E}(s_i)$) by verifying V. If a third-party attacker adds any modifications to the password-protected credentials $\{C_i\}'_i$, the user u_i aborts the protocol due to the invalidity of V (i.e., $V \neq \mathsf{MAC}(N_A, Y, X, \{C_i\}'_i)$) without sending out $\mathsf{Res}(\Pi_{R-BBS})$ to the server. Therefore, the attacker can not specify the user u_i in the attack of Sect. 4.

As it is clear, this countermeasure can also be used for the extended scheme (i.e., Sect. 4.3 of [19]) to support membership withdrawal. However, this countermeasure is not valid to the attack of a malicious server in [16].

Acknowledgments

We sincerely appreciate the anonymous reviewer's valuable comments on this paper. This work was partly supported by JST, Infrastructure Development for Promoting International S&T Cooperation and JSPS KAKENHI Grant Number JP16H02834.

References

[1] M.H. Au, W. Susilo, and Y. Mu, "Constant-Size Dynamic *k*-TAA," In Proc. of SCN 2006, LNCS 4116, vol.4116, pp.111–125, Springer-Verlag, 2006.
[2] D.J. Bernstein, "The Poly1305-AES Message-Authentication Code," In Proc. of FSE 2005, LNCS, vol.3557, pp.32–49, Springer-Verlage, 2005.
[3] D. Boneh, X. Boyen, and H. Shacham, "Short Group Signatures," In Proc. of CRYPTO 2004, LNCS, vol.3152, pp.41–55, Springer-Verlag, 2004.
[4] M. Bellare, R. Canetti, and H. Krawczyk, "Keying Hash Functions for Message Authentication," In Proc. of CRYPTO'96, LNCS, vol.1109, pp.1–15, Springer-Verlage, 1996.
[5] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, and P. Rogaway, "UMAC: Fast and Secure Message Authentication," In Proc. of CRYPTO'99, LNCS, vol.1666, pp.216–233, Springer-Verlage, 1999.
[6] J. Camenisch and A. Lysyanskaya, "A Signature Scheme with Efficient Protocols," In Proc. of SCN 2002, LNCS, vol.2576, pp.268–289, Springer-Verlage, 2003.
[7] FIPS PUB 198-1, "The Keyed-Hash Message Authentication Code (HMAC)," NIST, July 2008.
[8] T. Iwata and K. Kurosawa, "OMAC: One-Key CBC MAC," In Proc. of FSE 2003, LNCS, vol.2887, pp.129–153, Springer-Verlage, 2003.
[9] ISO/IEC 20009-4, "Information technology – Security techniques – Anonymous entity authentication – Part 4: Mechanisms based on weak secrets," https://www.iso.org/standard/64288.html.
[10] J. Katz and Y. Lindell, "Introduction to Modern Cryptography," Second Edition, CRC Press, 2015.
[11] L. Nguyen, "Accumulators from Bilinear Pairings and Applications," In Proc. of CT-RSA 2005, LNCS, vol.3376, pp.275–292, Springer-Verlag, 2005.
[12] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," In Proc. of EUROCRYPT'99, LNCS 1592, pp.223–238, Springer-Verlage, 1999.
[13] RFC 4418, "UMAC: Message Authentication Code using Universal Hashing," Informational, IETF, March 2006.
[14] RFC 7539, "ChaCha20 and Poly1305 for IETF Protocols," Informational, IRTF, May 2015.
[15] S.-H. Shin, K. Kobara, and H. Imai, "Anonymous Password-Authenticated Key Exchange: New Construction and Its Extensions," IEICE Trans. Fundamentals, vol.E93-A, no.1, pp.102–115, 2010.
[16] S.-H. Shin and K. Kobara, "On Unlinkability of Password-Based Anonymous Authentication," IEICE Trans. Fundamentals, vol.E98-A, no.6, pp.1320–1324, 2015.
[17] J. Yang and Z. Zhang, "A New Anonymous Password-Based Authenticated Key Exchange Protocol," In Proc. of INDOCRYPT 2008, LNCS, vol.5365, pp.200–212, Springer-Verlag, 2008.
[18] Y. Yang, J. Zhou, J. Wong, and F. Bao, "A New Approach for Anonymous Password Authentication," In Proc. of 2009 Annual Computer Security Applications Conference (ACSAC 2009), pp.199–208, 2009.
[19] Y. Yang, J. Zhou, J.W. Wong, and F. Bao, "Towards Practical Anonymous Password Authentication," In Proc. of 2010 Annual Computer Security Applications Conference (ACSAC 2010), pp.59–68, 2010.