Anonymous k-Show Credentials

Mohamed Layouni and Hans Vangheluwe

School of Computer Science, McGill University,
3480 University Street, Montreal, H3A 2A7, Quebec, Canada

Abstract. Privacy-preserving digital credentials are cryptographic tools that allow a user to prove a predicate about his/her identity or qualifications, without the verifying party learning additional information beyond the status of that predicate. The Identity Mixer (Idemix) [CL01] is a framework providing such credentials. In Idemix, we can distinguish two types of credentials: (1) one-time show credentials which can be shown only once before unveiling the identity of their holder, and (2) multi-show credentials which can be shown infinitely many times without the showings being linked to each other, or to the identity of their holder. In this paper, we bridge the gap between the two previous types of credentials, and extend Idemix to k-show credentials (for $k > 1$.) The k-show credentials we propose can be shown anonymously, but linkably, up to k times.

Keywords: Privacy-preserving digital credentials, anonymity, multiple-show credentials.

1 Introduction

With the increasing digitization of society, and the continuous migration of day-to-day services from the paper world to the digital world, digital credentials have become a very important tool. Similar to their paper counterparts, digital credentials are special documents, issued by a certification authority, that may contain a variety of information about their holder (e.g., identity attributes, qualifications, privileges, etc.) In addition, digital credentials have attractive features, that make them superior to their paper counterparts, such as searchability, large-scale data-mining, and knowledge discovery, just to name a few. With the latter features, comes also the disadvantage that credential holders are now a lot easier to monitor, and to have their privacy violated. Furthermore, digital credentials – by their very nature – are easy to clone and copy, and using them without proper safeguards could lead to serious security problems. To address this set of conflicting requirements, namely privacy and security requirements, privacy-preserving credentials have been invented [Cha85, CP92, Bra94, Bra00, CL01, CL04]. In a privacy-preserving digital credential system, one can generally distinguish three types of players: a certification authority, a user, and a verifier. In some cases, the certification authority and the verifier are controlled by the same entity. The certification authority issues a credential to a user who fulfills certain conditions. In exchange for goods and services, the user may be required to prove,

J. Lopez, P. Samarati, and J.L. Ferrer (Eds.): EuroPKI 2007, LNCS 4582, pp. 181–192, 2007.

to a service provider (the verifier), possession of a valid credential from the certification authority. The user may also be required to prove a predicate on the attributes encoded in his credential. The service provider may later decide to deposit a transcript of the interaction it had with the user, to the certification authority. The main requirements the credential system should satisfy are: (1) Non-forgeability: the user should not be able to succeed in proving the validity of forged credentials, or in proving predicates that are not satisfied by the attributes encoded on his CA-issued credential, and (2) Privacy: the verifier should not be able to learn any information about the user's credentials beyond what can be naturally inferred from the status of the proven predicate. The latter requirement can be refined even further, by adding constraints on the number of times a credential can be used. Based on this last criterion, we can distinguish three types of credentials:

1. Multiple-show credentials: they can be shown infinitely-many times without the showings being linked to each other, or to the issuing protocol instance where they were generated.
2. One-show credentials: they can be shown anonymously only once, before the identity of their holder is unveiled.
3. Limited- or k-show credentials, for ($k > 1$): they can be shown anonymously up to k times, after which the identity of the holder is revealed.

Privacy-preserving credential systems are becoming increasingly popular, and there is a growing interest in concrete implementations [Ide07, UPr07, Hig07]. The Identity Mixer [Ide07] is based on Camenisch and Lysyanskaya's credentials [CL01], and is one of today's most complete credential systems. Idemix provides a framework supporting only the first two types of credentials, namely multi-show credentials, and one-time show credentials.

Our Contribution: In this paper, we bridge the gap between the two first types of credentials, and extend the Idemix framework to k-show credentials (for $k > 1$.) A naive way to construct k-show credentials is by issuing k separate copies of one-show credentials, but this option obviously lacks efficiency. The solution we propose in this paper extends the one-time show credentials of [CL01] to k-show credentials without a significant increase in complexity. Compared to the protocols of [CL01], we only add 2 extra exponentiations and 1 proof of discrete logarithm knowledge to the user in the pseudonym creation protocol. For the issuing protocol, the user performs 3 more exponentiantions and a proof of knowledge for each additional showing allowed. Finally, the complexity of the showing protocol can be made very close to that of one-time show credentials [CL01] by using precomputations and fast exponentitation methods [Gor98].

Anonymous k-show credentials may be used in a variety of applications. They can be used for instance to build public transit passes, where a user is allowed to make up to k rides anonymously, after which the pass serial number will be uncovered, revoked, and added to a black list. In order to count the number of times a credential is shown, the issuing organization is able to link different showings of the same credential to each other, but not to the identity of the

credential holder, or for that matter, to the instance of the issuing protocol that generated the credential. This linking feature can also be found in the one-show credentials of [CL01] and [Bra94] where issuers rely on it to detect double-spending. The work in [LTW05] follows the same principle to recover lost electronic cash.

The remainder of this paper is organized as follows. In section 2 we give an overview of the basic credential system of [CL01], as well as the general setting. In section 3 we present our extension to k-show credentials. In section 4, we position our contribution with respect to related work. We conclude in section 5.

2 Review of the Idemix Credential System

2.1 General Setting

There are two main types of players in the Idemix system: users and organizations. Organizations offer both material (e.g., goods) and non-material (e.g., assertions) services to users. To provide those services, organizations require users to fulfill certain conditions. These conditions may include paying a fee, fulfilling a predicate about one's identity, or proving possession of an authorization from some recognized authority. More generally, this process consists in showing one or a set of credentials.

Users in turn want to benefit from those services without revealing unnecessary information about their identity. To remain anonymous, each user U possesses a different pseudonym $N_{(U,O_i)}$ with each organization O_i. For example, if an organization O_1 requires a user U to show that he has a valid credential from organization O_2, then U should be able to do so without O_1 and O_2 being capable of linking his pseudonyms $N_{(U,O_1)}$ and $N_{(U,O_2)}$. This property is called *unlinkability*. It should also be possible for user U to show a credential he obtained from organization O without O being able to retrace that credential back to the protocol instance where the credential was issued. This property is called *untraceability*.

In the Idemix system [CL01], a user U first registers with an organization O and obtains a pseudonym $N_{(U,O)}$, and a validating tag $P_{(U,O)}$ on it. The validating tag allows the user to prove that he actually owns the pseudonym. Next, user U obtains a credential from organization O. The credential is a pair denoted $(c_{(U,O)}, e_{(U,O)})$, such that $c_{(U,O)}^{e_{(U,O)}} = P_{(U,O)} d_O$, where d_O is a system parameter. Later, user U can prove to a verifying organization V that he holds a valid credential from organization O without either revealing the credential $(c_{(U,O)}, e_{(U,O)})$ or his pseudonym $N_{(U,O)}$ with organization O.

The Idemix framework is based on the strong RSA assumption and the decisions Diffie-Hellman assumption modulo a safe prime product.

In the following we begin with an overview of the system parameters. Then we give a brief description of the pseudonym generation protocol, the credential issuing protocol, and the credential showing protocol.

2.2 System Parameters and Notations

All RSA moduli used in the system are of length ℓ_n. Let the intervals $\Gamma =]-2^{\ell_\Gamma}, 2^{\ell_\Gamma}[$, $\Delta =]-2^{\ell_\Delta}, 2^{\ell_\Delta}[$, and $\Lambda =]-2^{\ell_\Lambda}, 2^{\ell_\Lambda+\ell_\Sigma}[$ be such that $\ell_\Delta = \epsilon(\ell_\Lambda + \ell_n) + 1$, where $\epsilon > 1$ is a security parameter, and $\ell_\Lambda > \ell_\Sigma + \ell_\Delta + 4$. Each organization O chooses a safe prime product modulus n_O and keeps the factorization secret. It also chooses random elements $a_O, b_O, d_O, g_O, h_O, v_O, z_O \in QR_{n_O}$ and publishes them along with n_O. The parameter ℓ_Λ is chosen such that computing discrete logarithms in QR_{n_O} with ℓ_Λ-bits exponents is hard.

In the remainder of this paper, $PK\{(\alpha, \beta, \cdots) : \mathcal{P}(A, B, \cdots; \alpha, \beta, \cdots)\}$, denotes, for public parameters $A, B, \cdots$, a proof of knowledge of secrets $\alpha, \beta, \cdots$, for which the public predicate $\mathcal{P}(\cdots)$ is satisfied.

2.3 Establishing a Pseudonym with an Organization

Let U be a user who wants to establish a pseudonym with organization O. Let $x_u \in \Gamma$ be U's master secret key (or ID). The protocol shown in figure 1 allows U to obtain a pseudonym $N_{(U,O)}$ and a validating tag $P_{(U,O)}$ such that $P_{(U,O)} = a_O^{x_u} b_O^{s_{(U,O)}}$, where $s_{(U,O)}$ is jointly chosen by U and O. Organization O learns nothing about the values of x_u and $s_{(U,O)}$.

2.4 Obtaining a Credential from an Organization

Organization O can issue a credential to user U who proves ownership of a previously established pseudonym and validating tag $(N_{(U,O)}, P_{(U,O)})$. The credential is a pair $(c_{(U,O)}, e_{(U,O)}) \in \mathbb{Z}_{n_O}^* \times \Lambda$ such that $P_{(U,O)} d_O = c_{(U,O)}^{e_{(U,O)}}$. Figure 2 describes how the issuing protocol works.

2.5 Showing a Credential to an Organization

User U can prove to any verifying organization V that he possesses a valid credential from organization O. In this proof V does not learn anything about the user's pseudonym with O, the credential he holds from O, or the secrets underlying that credential. Figure 3 explains how the show protocol works for one-time show credentials. Showing multiple-show credentials is simpler and can be easily derived.

If user U shows his one-time show credential more than once, organization O will have two challenge-response pairs (c_1, r_1) and (c_2, r_2), from which it can retrieve x_u and $s_{(U,O)}$, which in turn will determine $P_{(U,O)}$ and identify U.

3 Extension to *k*-Show Credentials

In this section we present an extension of the credential system described above. In addition to the multiple-show and one-show modes, where credentials can be spent either infinitely many times or only one time, we now present k-show credentials, with $k > 1$. k-show credentials can be spent k times without being

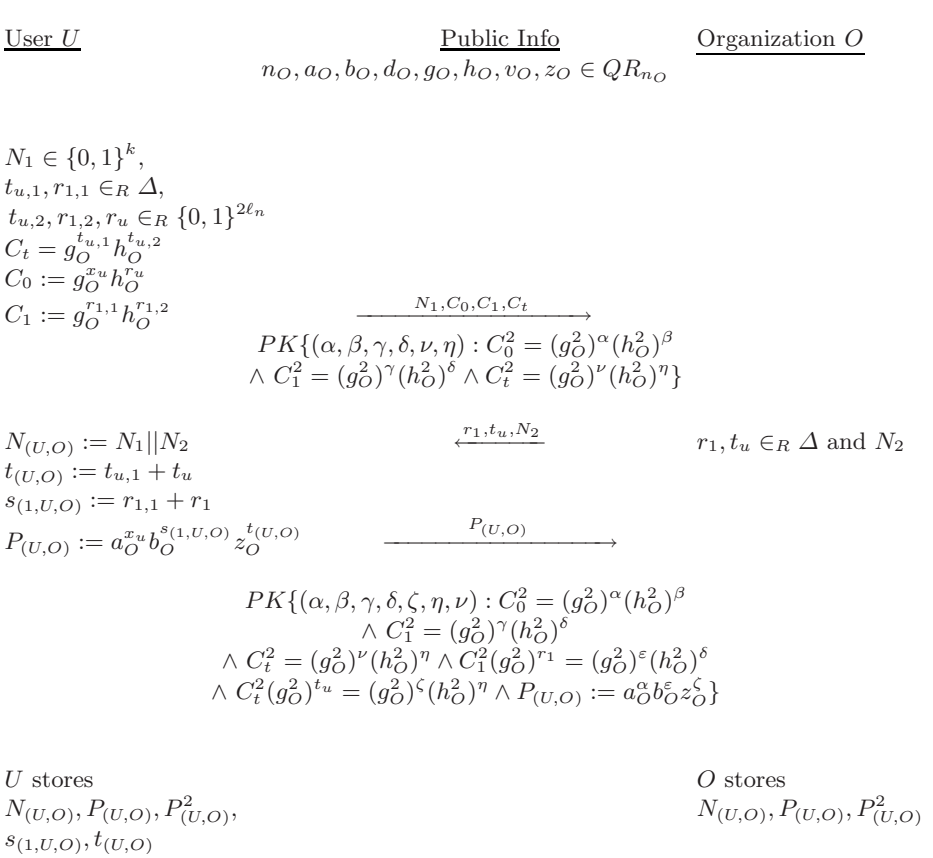

Fig. 1. Idemix pseudonym creation protocol

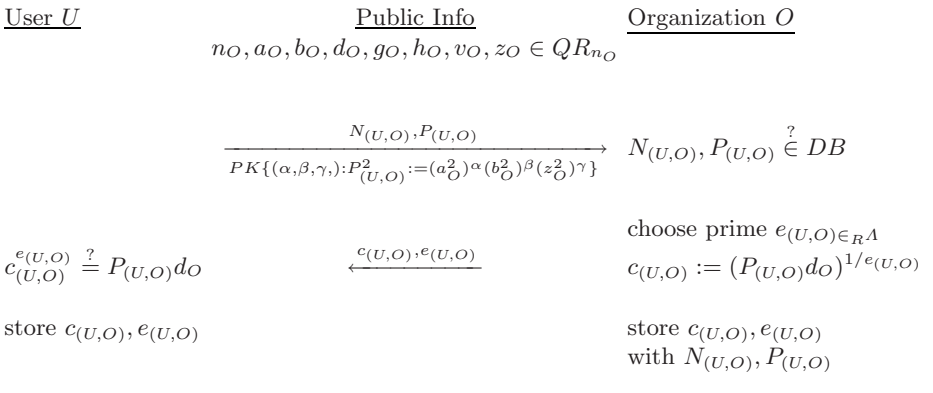

Fig. 2. Idemix credential issuing protocol

linked to the instance of the issuing protocol that generated them or to the identity of their owner. The identity of the owner is revealed only if the credential is spent more than k times.

User U	Public Info	Organization O
	$n_O, a_O, b_O, d_O, g_O, h_O, v_O, z_O \in QR_{n_O}$	
choose $r_1, r_2 \in_R \{0,1\}^{2\ell_n}$		
$A := c_{(U,O)} h_O^{r_1}$		
$B := h_O^{r_1} g_O^{r_2}$		
$H_{(u,O)} := h_O^{t_{(U,O)}}$	$\xrightarrow{A,B,H_{(u,O)}}$	
	$\xleftarrow{c}$	choose $c \in_R \{0,1\}^{\ell_c}$
$r := cx_u + s_{(U,O)}$	$\xrightarrow{r}$	
mark credential as void	$PK\{(\alpha,\beta,\gamma,\delta,\zeta,\xi,\varepsilon,\varphi) : H_{(u,O)} = h_O^{\varphi}$	
	$\wedge\ d_O^2 = (A^2)^{\alpha}(\frac{1}{a_O^2})^{\beta}(\frac{1}{b_O^2})^{\gamma}(\frac{1}{z_O^2})^{\varphi}(\frac{1}{h_O^2})^{\delta}$	
	$\wedge\ B^2 = (h_O^2)^{\varepsilon}(g_O^2)^{\zeta} \wedge 1 = (B^2)^{\alpha}(\frac{1}{h_O^2})^{\delta}(\frac{1}{g_O^2})^{\xi}$	If accepted
	$\wedge\ g_O^r = g_O^{\gamma}(g_O^c)^{\beta}$	store transcript
	$\wedge\ \beta \in \Gamma \wedge \gamma \in \Delta \wedge \varphi \in \Delta \wedge \alpha \in \Lambda\}$	for later checkings

Fig. 3. Idemix credential showing protocol (one-time show mode)

For the purpose of credential revocation, an additional party will be added to the setting. This party is an independent outsider to the system, and is denoted CA. When showing a credential, a user verifiably encrypts a piece of identifying information under the CA's public key. The encryption also specifies the condition under which the ciphertext is decrypted. This is achieved using the Cramer-Shoup encryption scheme [CS98].

Additional system parameters. These are parameters to be used in the verifiable encryption scheme. The CA chooses a group $G = \langle g \rangle = \langle h \rangle$ of prime order $q > 2^{\ell_\Gamma}$. Then he chooses $x_1, x_2, x_3, x_4, x_5 \in_R \mathbb{Z}_q$ and keeps them secret. The CA then computes the tuple $(y_1, y_2, y_3) := (g^{x_1}h^{x_2}, g^{x_3}h^{x_4}, g^{x_5})$ and publishes it as his public key.

3.1 Establishing a Pseudonym with an Organization for Obtaining Revocable Credentials

This protocol can be used not only for k-show credentials but for multiple-show credentials as well. It simply gives an organization the necessary information it could use to identify a user and revoke his credentials in case a certain revocation condition is fulfilled. The protocol is shown on figure 4.

Several variables are used in this protocol. Following are the roles played by the most important ones.

- $t_{(U,O)}$ which is only known to the user will be used to link the showings of the same k-show credential to each other.
- x_u is used to identify the user globally by the CA. x_u is only known to the user.

User U	Public Info	Organization O
	$n_O, a_O, b_O, d_O, g_O, h_O, v_O, z_O \in QR_{n_O}$	
		G, g, h, y_1, y_2, y_3
$N_1 \in \{0,1\}^k$,		
$x_u, x_{(u,O)} \in \Gamma$		
$t_{u,1}, r_{1,1} \in_R \Delta$,		
$t_{u,2}, r_{1,2}, r_u \in_R \{0,1\}^{2\ell_n}$		
$C_t = g_O^{t_{u,1}} h_O^{t_{u,2}}$		
$C_0 := g_O^{x_u} h_O^{r_u}$		
$C_1 := g_O^{r_{1,1}} h_O^{r_{1,2}}$	$\xrightarrow{N_1, C_0, C_1, C_t}$	
	$PK\{(\alpha,\beta,\gamma,\delta,\nu,\eta) : C_0^2 = (g_O^2)^\alpha (h_O^2)^\beta$	
	$\wedge\, C_1^2 = (g_O^2)^\gamma (h_O^2)^\delta \wedge C_t^2 = (g_O^2)^\nu (h_O^2)^\eta\}$	
	$\xleftarrow{r_1, t_u, N_2}$	$r_1, t_u \in_R \Delta$ and N_2
$N_{(U,O)} := N_1 \| N_2$		
$t_{(U,O)} := t_{u,1} + t_u$		
$s_{(1,U,O)} := r_{1,1} + r_1$		
$P_{(U,O)} := a_O^{x_u} b_O^{s_{(1,U,O)}} .z_O^{t_{(U,O)}} v_O^{x_{(u,O)}}$		
If $O =$ CA then		
$Y_u := g^{x_u}$	$\xrightarrow{P_{(U,O)}, Y_u}$	
	$PK\{(\alpha,\beta,\gamma,\delta,\zeta,\eta,\nu,\xi) : C_0^2 = (g_O^2)^\alpha (h_O^2)^\beta$	
	$\wedge\, C_1^2 = (g_O^2)^\gamma (h_O^2)^\delta \wedge C_1^2 (g_O^2)^{r_1} = (g_O^2)^\varepsilon (h_O^2)^\delta$	
	$\wedge\, C_t^2 = (g_O^2)^\nu (h_O^2)^\eta \wedge\ C_t^2 (g_O^2)^{t_u} = (g_O^2)^\zeta (h_O^2)^\eta$	
	$\wedge\, P_{(U,O)} := a_O^\alpha b_O^\varepsilon z_O^\zeta v_O^\xi \wedge Y_u = g^\alpha\}$	
else		
$Y_{(u,O)} := g^{x_{(u,O)}}$	$\xrightarrow{P_{(U,O)}, Y_{(u,O)}}$	
	$PK\{(\alpha,\beta,\gamma,\delta,\zeta,\eta,\nu,\xi) : C_0^2 = (g_O^2)^\alpha (h_O^2)^\beta$	
	$\wedge\, C_1^2 = (g_O^2)^\gamma (h_O^2)^\delta \wedge C_1^2 (g_O^2)^{r_1} = (g_O^2)^\varepsilon (h_O^2)^\delta$	
	$\wedge\, C_t^2 = (g_O^2)^\nu (h_O^2)^\eta \wedge\ C_t^2 (g_O^2)^{t_u} = (g_O^2)^\zeta (h_O^2)^\eta$	
	$\wedge\, P_{(U,O)} := a_O^\alpha b_O^\varepsilon z_O^\zeta v_O^\xi \wedge Y_{(u,O)} = g^\xi\}$	
U stores		O stores
$N_{(U,O)}, P_{(U,O)}, P_{(U,O)}^2$,		$N_{(U,O)}, P_{(U,O)}, P_{(U,O)}^2$,
$s_{(1,U,O)}, t_{(U,O)}$,		Y_u or $Y_{(u,O)}$
Y_u or $Y_{(u,O)}$		

Fig. 4. Pseudonym creation protocol for revocable k-show credentials

- $x_{(U,O)}$ will be used to identify the user locally at the issuing organization. $x_{(U,O)}$ is only known to the user.
- Y_u can be used only when a user violates the agreement governing the usage of his credential (e.g., overshowing). Y_u allows the CA to reveal a user's identity.
- $Y_{(u,O)}$ is similar to Y_u, but de-anonymizes the user only locally by revealing his pseudonym to the issuing organization O.

3.2 Obtaining a *k*-Show Revocable Credential from an Organization

User U executes the protocol of Figure 5 to obtain a k-show credential with organization O. The tuple $(P_{(U,O)}, Q_{(k,U,O)}, c_{(k,U,O)}, e_{(k,U,O)})$ is U's k-show credential with organization O. The pseudonym $P_{(U,O)}$ does not depend on k and can therefore be used both in the multiple-show as well as the k-show setting.

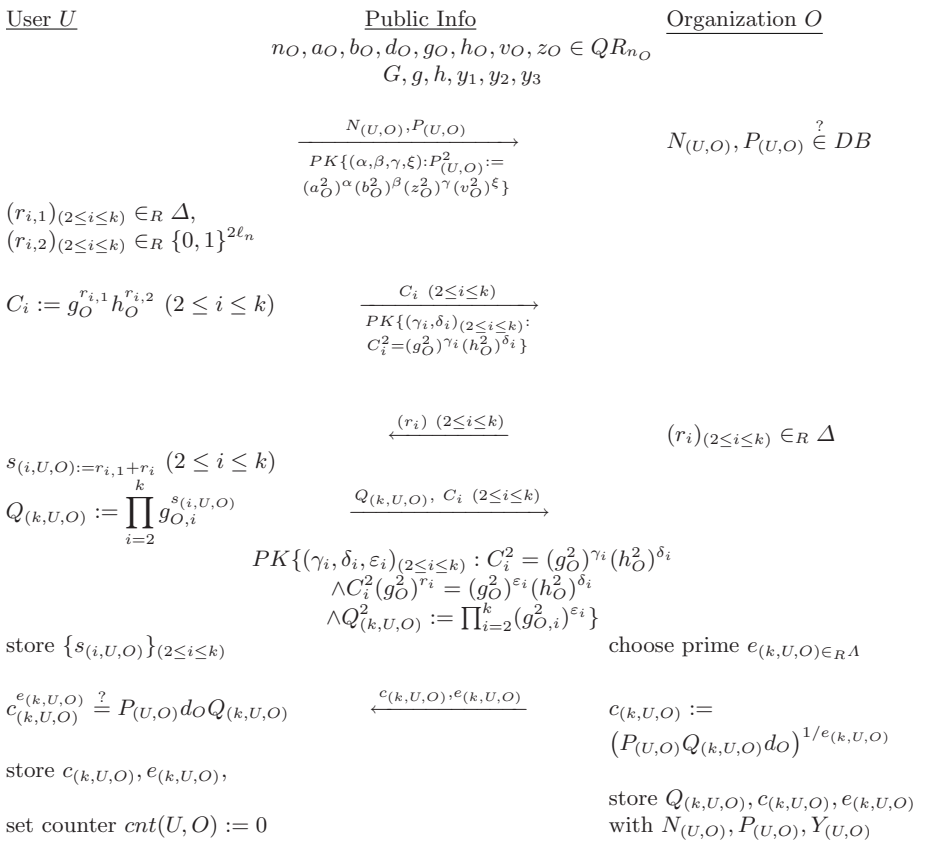

Fig. 5. Issuing protocol for revocable k-show credentials

3.3 Showing a Revocable *k*-Show Credential to an Organization

User U is allowed to show his credential up to k times without the showings being linked to his identity or to the instance of the issuing protocol by which it has obtained the credential. Figure 6 describes how the showing protocol works.

3.4 Local User Identification and Credential Revocation

A verifying organization V in the showing protocol above, submits the showing transcript offline to the issuing organization O, which in turn will be able to

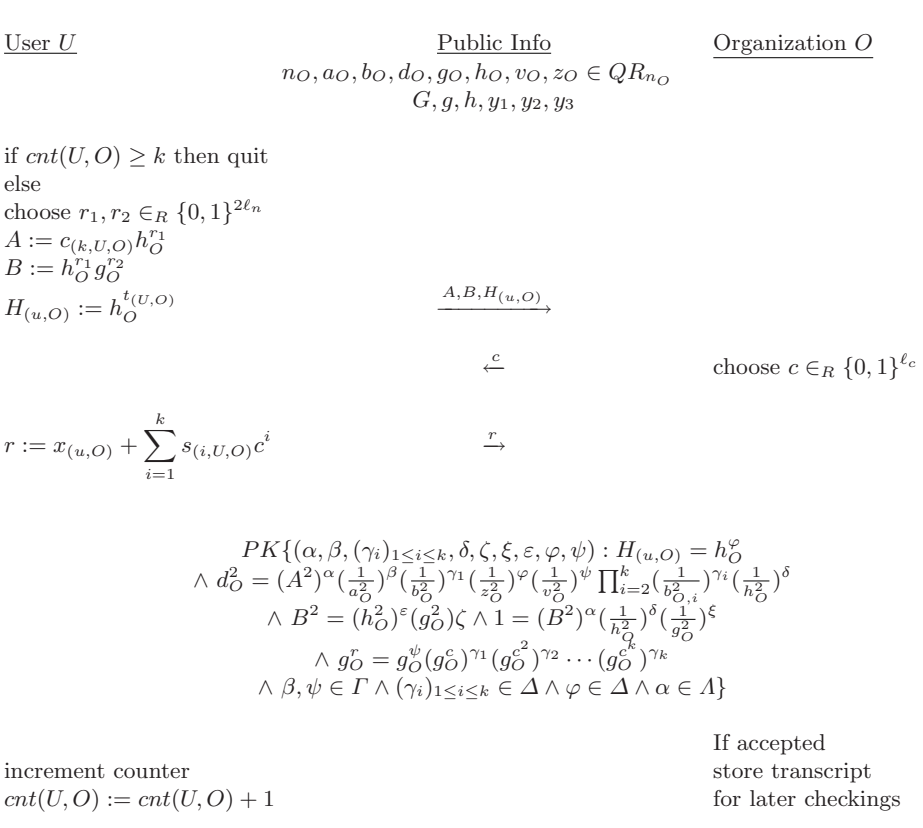

Fig. 6. Showing protocol for revocable k-show credentials

link the user's showings via $H_{(u,O)}$ without actually identifying his pseudonym. In case user U shows his credential beyond the allowed threshold k, then any party can verifiably recover his secret key $x_{(u,O)}$ by interpolating (in the sense of Lagrange) any subset of $(k+1)$ challenge-response pairs (c, r) that were used in the showing protocols conducted by the user. Given $x_{(u,O)}$, organization O computes $Y_{(u,O)} = g^{x_{(u,O)}}$, which will determine user U's pseudonym $N_{(U,O)}$, and enable the revocation of his credentials. The value of $H_{(u,O)}$, retrieved from the incriminating transcripts above, is added to a blacklist of revoked credentials, and used to detect subsequent attempts to show revoked credentials.

3.5 Global User Identification and Credential Revocation

Global revocation allows for the identification of a user by an external referee to the system, denoted CA. The conditions leading to the de-anonymization of the user are negotiated by the verifying organization and the user prior to executing the show protocol. Once the de-anonymization condition is agreed upon, the user may proceed with the showing protocol as follows. The user starts by verifiably encrypting a part of his identity (Y_u) using the CA's public key and sends the

ciphertext, along with a proof of correctness, to the verifying organization. The de-anonymization condition is tied to the encryption, thereby forcing the CA to only decrypt the ciphertext when this condition is met. We denote this de-anonymization condition by L. The verifiable encryption scheme of Cramer and Shoup [CS98] is used to achieve this.

To show a credential to verifying organization V, user U starts by executing the protocol of Figure 7 with V. Both U and V then proceed with the rest of the show protocol as shown in Figure 6.

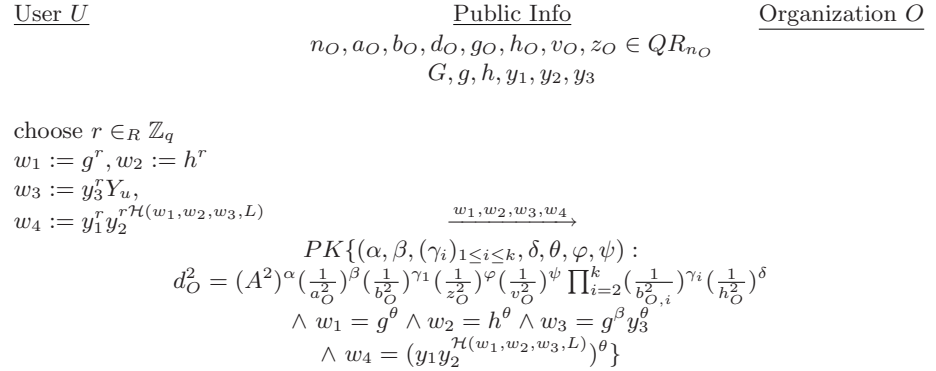

Fig. 7. Verifiable encryption of a user's revocation information

Global revocation. When provided with an encryption (w_1, w_2, w_3, w_4), a de-anonymization condition L, and evidence that L is satisfied, the CA checks whether $w_4 \stackrel{?}{=} w_1^{x_1+x_3\mathcal{H}(w_1,w_2,w_3,L)} w_2^{x_2+x_4\mathcal{H}(w_1,w_2,w_3,L)}$. If this is the case, the CA computes $Y_u = w_3/w_1^{x_5}$ which will determine the identity of user U.

4 Related Work

There have been a number of research efforts on limited-show credentials in the literature. In the following we list the ones that are most relevant to our work. In [LTW05], Lui et al., go around the problem of k-show credentials, by making a user initially fill-up a "wallet" of k coins, each of which can be shown only once. The identity of the user will be revealed as soon as a given coin is shown a second time. The work in [LTW05], also provides a mechanism to recognize and recover lost coins belonging to a given user. In [TFS04, NSN05], the authors propose constructions allowing users to show their credentials, anonymously and unlinkably, up to k times. All additional shows will be linked to a subset of the initial k shows, but the system falls short of identifying the abusers. More recently, Camenisch et al. [CHK$^+$06] proposed a credential system that allows a user to anonymously authenticate at most k times within a pre-defined time period. While the latter represents a significant step towards solving the problem of k-show credentials, it still does not solve it, since users just need to wait until

the beginning of the next time period, before starting to reuse their credentials all the way from scratch.

5 Conclusion

This work extends the Identity Mixer framework to support anonymous k-show credentials, for $k > 1$. The proposed construction allows a user to show his credential up to k times, without the verifying or issuing organizations being able to link the different showings to the identity of the credential holder, or to the instance of the issuing protocol that generated the credential. Similar to the original framework, the credentials proposed in this work, are revocable both locally by the issuing organization, and globally by a trusted third party, upon fulfillment of a pre-defined de-anonymization condition.

Acknowledgement

This work is supported by the IWT SBO ADAPID project (Advanced Applications for e-ID cards in Flanders). Mohamed Layouni was partially funded by a doctoral scholarship from the Universitary Mission of Tunisia in North America.

References

[Bra94] Brands, S.: Untraceable off-line cash in wallet with observers. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 302–318. Springer, Heidelberg (1994)

[Bra00] Brands, S.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press, Cambridge (2000)

[Cha85] Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Commun. ACM 28(10), 1030–1044 (1985)

[CHK+06] Camenisch, J., Hohenberger, S., Kohlweiss, M., Lysyanskaya, A., Meyerovich, M.: How to win the clonewars: efficient periodic n-times anonymous authentication. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 201–210. ACM Press, New York (2006)

[CL01] Camenisch, J., Lysyanskaya, A.: Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001)

[CL04] Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)

[CP92] Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 89–105. Springer, Heidelberg (1993)

[CS98] Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)

[Gor98] Gordon, D.M.: A survey of fast exponentiation algorithms. Journal of Algorithms 27, 129–146 (1998)

[Hig07] The Higgins Trust Framework Project. URL functional as of (February 2007) http://www.eclipse.org/higgins/

[Ide07] The Identity Mixer: URL functional as of (February 2007), http://www.zurich.ibm.com/security/idemix/

[LTW05] Liu, J.K., Tsang, P.P., Wong, D.S.: Recoverable and untraceable e-cash. In: Chadwick, D., Zhao, G. (eds.) EuroPKI 2005. LNCS, vol. 3545, pp. 206–214. Springer, Heidelberg (2005)

[NSN05] Nguyen, L., Safavi-Naini, R.: Dynamic k-times anonymous authentication. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 318–333. Springer, Heidelberg (2005)

[TFS04] Teranishi, I., Furukawa, J., Sako, K.: k-times anonymous authentication (extended abstract). In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 308–322. Springer, Heidelberg (2004)

[UPr07] The U-Prove SDK: URL functional as of (February 2007), http://www.credentica.com/uprove_sdk.html