Untraceability and Profiling are Not Mutually Exclusive*

Sébastien Canard[1] and Amandine Jambert[1,2]

[1] Orange Labs, 42 rue des Coutures, BP6243, 14066 Caen Cedex, France
[2] IMB, Université Bordeaux 1, 351 cours de la Libération, 33405 Talence, France

Abstract. In this paper, we study the concept of privacy-preserving multi-service subscription systems. With such system, service providers can propose to their customers, by the way of a subscription, several distinct services that users can access while being anonymous. We moreover study how users can be untraceable *w.r.t.* the service provider during the subscription process, in such a way that it is additionally possible to make profiling on the users' customs. This permits the service provider to propose some advertisements to users while protecting the privacy of the latter, even this may be seen as contradictory. We also propose concrete instantiations, based on signature schemes with extensions from Camenisch and Lysyanskaya.

1 Introduction

Nowadays, more and more services are available on the internet. Some of them are free but, some others imply a payment from the customers. Users may pay each time they use the provided service, or subscribe to this service to use it once [12, 8], a fixed number of time, or each time they want during a fixed time period [2]. In this paper, we focus on the latter case: a user subscribes to a service (or a set of services) and can use it as she wants. More precisely, we focus on the case where service providers propose to their customers several distinct services for which it is necessary to subscribe before using them.

Such subscription should not be done to the detriment of privacy principles and users may not want to be traced in their actions. It should be possible for a user to be anonymous and untraceable when she access a subscribed service, as described by Blanton in [2], or in [18]. It is also possible to do better than the Blanton system by additionally making the user anonymous and untraceable *w.r.t.* the service provider during the subscription process. Note that in this case, it is necessary to add a privacy-preserving payment system such as Secure Electronic Transaction (SET) [16], e-cash [5] or multi-coupon [8] systems. In the following, we only focus on the subscription part and do not treat this payment phase.

* This work has been financially supported by the French Agence Nationale de la Recherche and the TES Cluster under the PACE project.

Appeared in S. K. Katsikas, J. Lopez and M. Soriano (Eds.): TrustBus 2010, volume 6264 of LNCS, pp. 117-128, 2010.

In this paper, we also study "profiling", that is the analysis of a group of customers to determine what characteristics they might have in common. This permits a service provider to know what set of services one user is interested in, such that this service provider is able to put some well-chosen advertisements for a particular user in a personalized web page, influencing this user to buy some new services, according to her preferences.

The untraceability of a user during a subscription or an access to some services may be considered as contradictory with the possibility for the service provider to make such profiling. In this paper, we show that this is not true. We thus study different levels of untraceability during use and/or subscription in order to allow the service provider to make such profiling. More precisely, we propose different multi-service subscription schemes which permit to balance both untraceability and profiling during purchase, while keeping the user untraceable during the use of one service.

The paper is organized as follows. In the next section, we introduce the concept of multi-service subscription scheme. Then, we propose a new system based on signature schemes with extensions proposed by Camenisch and Lysyanskaya. Finally we introduce different extensions balancing untraceability and profiling before to conclude.

2 Multi-Service Subscription Systems

A multi-service subscription system is composed of two types of actors: users, denoted $\mathcal{U}$, who want to subscribe and use services provided by a service provider $\mathcal{SP}$. A service provider provides a set of f different services, each of them being identified by a unique identifier denoted s_i. Each user can use a specific service as soon as she subscribes to it. A user is known to be a subscriber by owning a subscription certificate. At any time, the user can subscribe to more services provided by the same (or not) service provider. Concerning privacy, the user is anonymous and untraceable when she uses a specific service. In the following, we more formally describe this concept.

2.1 Procedures

Formally speaking, a multi-service subscription system is composed of the following procedures, where λ is a security parameter.

- Setup is an algorithm executed by some designated entities which on input 1^λ outputs the parameters param of the system. These parameters can be common for several service providers.
- SPSetup is an algorithm executed by $\mathcal{SP}$ providing f different services to generates the set $\mathcal{S}$ of service identifiers $s_1, \cdots, s_f$, on input 1^λ and param. The service provider also outputs a pair of keys $(\mathsf{spsk}, \mathsf{sppk})$. The public key is certified by some designated authorities, for example using a PKI.
- USetup is a procedure which permits the user to obtain a pair of keys $(\mathsf{usk}, \mathsf{upk})$, upk being published. As for sppk, this public key may be certified.

Appeared in S. K. Katsikas, J. Lopez and M. Soriano (Eds.): TrustBus 2010, volume 6264 of LNCS, pp. 117-128, 2010.

– Subscribe is a protocol between $\mathcal{U}$ and $\mathcal{SP}$, in which $\mathcal{U}$ subscribes to some services. $\mathcal{U}$ gives to $\mathcal{SP}$ a subset $\mathcal{S}_\mathcal{U}$ of the set $\mathcal{S}$ of all provided services. $\mathcal{U}$ takes as input usk, upk, param, $\mathcal{S}_\mathcal{U}$, $\mathcal{S}$ and sppk and $\mathcal{SP}$ takes as input spsk, sppk, param and $\mathcal{S}$. The user outputs a subscription certificate cert.
– AddSubscribe permits a user $\mathcal{U}$ owning a certificate cert to subscribe to new services and thus to update cert so that it incorporates the new services. More formally, this is a protocol between $\mathcal{U}$, taking on input usk, param, $\mathcal{S}_\mathcal{U}$, $\mathcal{S}$, sppk and the initial certificate cert, and $\mathcal{SP}$, taking on input spsk, param, $\mathcal{S}$ and sppk. The user outputs an updated subscription certificate $\widetilde{\mathsf{cert}}$ which corresponds to her subscription to a subset $\tilde{\mathcal{S}}_\mathcal{U} \subset \mathcal{S}$ such that $\mathcal{S}_\mathcal{U} \subset \tilde{\mathcal{S}}_\mathcal{U}$.
– Use permits to $\mathcal{U}$ to prove to $\mathcal{SP}$ that she has the right to use a service $s \in \mathcal{S}$. The user takes on input cert, usk, param, the service $s \in \mathcal{S}_\mathcal{U} \subset \mathcal{S}$ and sppk, while the service provider uses spsk, param, $\mathcal{S}$ and sppk. The output of this protocol is either 1 if the user has the right to obtain the service s or 0.

2.2 Security and Efficiency Issues

There are several security and efficiency issues in our context, which ones are based on the work from [2].

– **Correctness**: any subscriber can use, thanks to the Use protocol with $\mathcal{SP}$, the services she subscribed thanks to the Subscribe or the AddSubscribe procedure with that $\mathcal{SP}$.
– **Soundness**: even a coalition of legitimate users is unable to obtain access to non-subscribed services. The aim of an adversary, who may play several users, is to be accepted during a Use on a service s_i while having played no Subscribe or AddSubscribe protocol on that service with this $\mathcal{SP}$.
– **Anonymity**: even the $\mathcal{SP}$ is unable to identify a user within legitimate users or to decide whether two executions of Use come from the same user. An adversary, playing the role of $\mathcal{SP}$, should be unable to decide between two chosen users which one is playing a Use with the fraudulent $\mathcal{SP}$.
– **Compactness**: the size of the certificate cert should not depend on the number of embedded services.

2.3 Profiling Definition

One of the aim of the $\mathcal{SP}$ is to profile costumers, *i.e.* to analyse his group of customers to determine what characteristics they have in common. This is used by $\mathcal{SP}$ to better direct their future sales and marketing programs. Unfortunately, this may interfere with privacy.

3 Our Basic Construction: Scheme 1

3.1 Notation and Building Blocks

In the following, a bilinear environment is denoted $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ where p is a prime number, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are two groups of order p, g_1 (resp. g_2) is a generator of $\mathbb{G}_1$ (resp. $\mathbb{G}_2$) and $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ is a bilinear map.

Appeared in S. K. Katsikas, J. Lopez and M. Soriano (Eds.): TrustBus 2010, volume 6264 of LNCS, pp. 117-128, 2010.

Zero-Knowledge Proof of Knowledge. Roughly speaking, a Zero Knowledge Proof of Knowledge (ZKPK) is an interactive protocol during which a prover $\mathcal{P}$ proves to a verifier $\mathcal{V}$ that she knows a set of secret values verifying a given relation without revealing anything else[3]. In the following, we denote by $\text{POK}(\alpha_1, \ldots, \alpha_q : \mathsf{R}(\alpha_1, \ldots, \alpha_q))$ a proof of knowledge of the secrets $\alpha_1, \ldots, \alpha_q$ verifying the relation R. In this paper, we only consider the case where secrets are discrete logarithms in relations constructed over a group of prime order: proof of knowledge of a discrete logarithm [19] $\text{POK}(\alpha : y = g^\alpha)$; proof of knowledge of a representation [17] $\text{POK}(\alpha_1, \ldots, \alpha_q : y = g_1^{\alpha_1} \ldots g_q^{\alpha_q})$; and proof of equality of discrete logarithms [11] $\text{POK}(\alpha : y = g^\alpha \wedge z = h^\alpha)$. Such proofs of knowledge can be turned to non-interactive proofs of knowledge (*a.k.a.* signatures of knowledge) by using the Fiat-Shamir heuristic [14].

Signature schemes with extensions. The concept of signature schemes with extensions was introduced by Camenisch and Lysyanskaya [6]. Such schemes are standard signature schemes with some additional features. The first additional feature is the possibility to sign a message (SIGN algorithm) which is decomposed into several blocks $m = m_0 \| \cdots \| m_\ell$. The second one is an algorithm, denoted CSIGN, which permits the signer to sign a commitment C on some unknown values $(m_0, \cdots, m_\ell)$, using the Pedersen commitment scheme. Finally, it is possible to prove the knowledge of a valid signature on a message divided into blocks without revealing the message nor the signature: $\text{POK}(m = m_0 \| \cdots \| m_\ell, \sigma : \text{VERIF}(m, \sigma, \mathsf{spk}) = 1)$.

It exists several constructions of such signature schemes with extensions [6, 7]. We here focus on the one [7] based on the q-SDH assumption and related to the BBS group signature scheme [3].

3.2 High Level Description of Scheme 1

In our basic solution, each service provided by $\mathcal{SP}$ is known by a specific identifier s_i and is related to one generator h_i and one scalar n_i which is used to state that this service has not been subscribed. $\mathcal{SP}$ can generate signatures with extensions to sign the subscribed services $(s_{i_1}, \cdots, s_{i_k})$ and the unsubscribed ones $(n_{i_{k+1}}, \cdots, n_{i_f})$, together with the secret key usk of the subscriber, so that only her can use this subscription. This is done during the SUBSCRIBE procedure by using an interactive signing protocol (see below).

The ADDSUBSCRIBE procedure consists in executing a new signing protocol to add messages to a signature with extensions. For this purpose, we improve signature schemes with extensions by adding a new feature, making possible to update a previously obtained signature to add sub messages. Finally, the USE protocol consists for the user in proving her knowledge of a signature with extension on the wanted service, without revealing the signature nor the other

[3] These protocols are also used to prove that some public values are well-formed from known secret ones (*e.g.* a ciphertext *w.r.t.* a known secret plaintext).

Appeared in S. K. Katsikas, J. Lopez and M. Soriano (Eds.): TrustBus 2010, volume 6264 of LNCS, pp. 117-128, 2010.

subscribed services. We now detailed each procedure one by one, using the q-SDH based signature scheme with extensions from [7, 3].

3.3 Setup Procedures

Let λ be a security parameter. The SETUP procedure consists in generating a bilinear environment $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$. Let $g, h \in \mathbb{G}_1$. The SPSETUP algorithm consists in choosing, for each service provider, the number f of proposed services[4]. Each service is next associated to three different values: one scalar, denoted s_i, to state that the service is subscribed, one another scalar, denoted n_i, to state that the service is unsubscribed, and one group element $h_i \in \mathbb{G}_1$. $\mathcal{SP}$ also generates at random the signature secret key γ of the chosen signature scheme with extension and publishes the corresponding public key $w = g_2^{\gamma}$ in the service provider public key sppk. Finally, $\mathcal{U}$ is related to a secret key usk and known by the public key $\mathsf{upk} = g^{\mathsf{usk}}$, which one may be certified by using a PKI. We will see other possibilities in Section 4.

3.4 Subscription Procedure

We suppose that $\mathcal{U}$, with the key pair $(\mathsf{usk}, \mathsf{upk})$, wants to subscribe to $k \leq f$ services identified by $s_{i_1}, \cdots, s_{i_k}$ where the i_j's belong to $[1, f]$. We denote by $\mathcal{I} = \{i_1, \cdots, i_k\} \subset [1, f]$. This protocol is an interactive protocol of the signature scheme with extensions between the user $\mathcal{U}$ and the signer $\mathcal{SP}$, which permits the user to obtain a signature on the $f + 1$ following committed values: usk, all s_j for $j \in \mathcal{I}$ and all n_j for $j \in [1, f] \setminus \mathcal{I}$. During this protocol, the value usk is added by the user while the service identifiers s_j and n_j are committed by the service provider[5]. More precisely, we have the following steps.

1. The user first commits to a secret s' and her user secret key usk: $C' = h^{s'} h_0^{\mathsf{usk}}$.
2. She produces U as a proof of knowledge that C' is well-formed using known s' and usk. Note that during this step, the user should prove, within the U proof of knowledge, that the committed value usk is related to the given public key upk: $U = \text{POK}(s', \mathsf{usk} : C' = h^{s'} h_0^{\mathsf{usk}} \wedge \mathsf{upk} = g^{\mathsf{usk}})$. The user sends to $\mathcal{SP}$ the commitment C', her public key upk (and, if needed, the X.509 certificate), the proof U and the wanted services $s_{i_1}, \cdots, s_{i_k}$.
3. $\mathcal{SP}$ adds to the commitment C' the values corresponding to the subscribed (the s_j's) and the unsubscribed (the n_j's) services, and modifies s' to $s = s' + s''$ in the new commitment $C = C' h^{s''} \prod_{j \in \mathcal{I}} h_j^{s_j} \prod_{j \in [1,f] \setminus \mathcal{I}} h_j^{n_j}$.
4. $\mathcal{SP}$ finally signs the commitment C so that $\mathcal{U}$ obtains a signature (A, x) on $(s, \mathsf{usk}, \{s_j\}_{j \in \mathcal{I}}, \{n_j\}_{j \in [1,f] \setminus \mathcal{I}})$. For this purpose, x is chosen at random in $\mathbb{Z}_p^*$ and A is computed as $A = (g_1 h^s h_0^{\mathsf{usk}} \prod_{j \in \mathcal{I}} h_j^{s_j} \prod_{j \in [1,f] \setminus \mathcal{I}} h_j^{n_j})^{\frac{1}{\gamma + x}}$. $\mathcal{SP}$ also saves the commitment C and the services $s_{i_1}, \cdots, s_{i_k}$ subscribed by $\mathcal{U}$. The subscription certificate cert is finally the signature with extension $\sigma = (A, x)$.

[4] This number can be updated by generating the corresponding triple (s_i, n_i, h_i).

[5] The values n_j are necessary to improve the untraceability, since one can learn some information on $\mathcal{U}$ by knowing that she has e.g. registered to only 3 services.

Appeared in S. K. Katsikas, J. Lopez and M. Soriano (Eds.): TrustBus 2010, volume 6264 of LNCS, pp. 117-128, 2010.

3.5 Addition of Services

We now suppose that $\mathcal{U}$ has previously subscribed to k services. Thus, she knows a certificate represented by the signature $\sigma = (A, x)$ on the message $(s, \mathsf{usk}, \{s_j\}_{j\in\mathcal{I}}, \{n_j\}_{j\in[1,f]\setminus\mathcal{I}})$ (also written $C = h^s h_0^{\mathsf{usk}} \prod_{j\in\mathcal{I}} h_j^{s_j} \prod_{j\in[1,f]\setminus\mathcal{I}} h_j^{n_j}$). $\mathcal{U}$ now wants to subscribe to l additional services (for simplicity, we denote $\ell = k + l$) identified by $s_{i_{k+1}}, \cdots, s_{i_\ell}$. In the following, we denote by $\tilde{\mathcal{I}} = \{i_{k+1}, \cdots, i_\ell\} \cup \mathcal{I}$. Our aim is to make one single certificate incorporating the previously obtained services and the new ones. More precisely, we have the following steps.

1. The user first sends to $\mathcal{SP}$ her public key upk and the previously signed message in the form C above. She finally produces the proof of knowledge
$$V = \textsc{Pok}(s, \mathsf{usk} : C / \prod_{j\in\mathcal{I}} h_j^{-s_j} \prod_{j\in[1,f]\setminus\mathcal{I}} h_j^{-n_j} = h^s h_0^{\mathsf{usk}} \wedge \mathsf{upk} = g^{\mathsf{usk}})$$
2. We consider that the aggregation of subscribed services is done by $\mathcal{SP}$ since it knows the identity of $\mathcal{U}$. $\mathcal{SP}$ retrieves in its database the value C and the services $s_{i_1}, \cdots, s_{i_k}$ already subscribed by $\mathcal{U}$, verifies U, adds to C the values $s_{i_{k+1}}, \cdots, s_{i_\ell}$ and modifies s to $\tilde{s} = s + \tilde{s}'$ in the new commitment $\tilde{C} = C h^{\tilde{s}'} \prod_{j\in\tilde{\mathcal{I}}\setminus\mathcal{I}} h_j^{s_j - n_j} = h^{\tilde{s}} h_0^{\mathsf{usk}} \prod_{j\in\tilde{\mathcal{I}}} h_j^{s_j} \prod_{j\in[1,f]\setminus\tilde{\mathcal{I}}} h_j^{n_j}$.
3. $\mathcal{SP}$ finally signs $\tilde{C}$ so that $\mathcal{U}$ obtains a signature $(\tilde{A}, \tilde{x})$ on $(s, \mathsf{usk}, \{s_j\}_{j\in\tilde{\mathcal{I}}}, \{n_j\}_{j\in[1,f]\setminus\tilde{\mathcal{I}}})$, that is such that $\tilde{A} = (g_1 h^{\tilde{s}} h_0^{\mathsf{usk}} \prod_{j\in\tilde{\mathcal{I}}} h_j^{s_j} \prod_{j\in[1,f]\setminus\tilde{\mathcal{I}}} h_j^{n_j})^{\frac{1}{\gamma+\tilde{x}}}$, where $x \in_R \mathbb{Z}_p^*$. $\mathcal{SP}$ should not take the same x used during Subscribe since it permits $\mathcal{U}$ to forge signatures. $\mathcal{SP}$ saves the new services $s_{i_{k+1}}, \cdots, s_{i_\ell}$ subscribed by $\mathcal{U}$. The new subscription certificate $\widetilde{\mathsf{cert}}$ is finally $\tilde{\sigma} = (\tilde{A}, \tilde{x})$.

3.6 The Use Protocol

We next imagine that $\mathcal{U}$, who has subscribed to services $s_{i_1}, \cdots, s_{i_\ell}$, wants to use *e.g.* s_{i_1}. The Use protocol is based on the ZKPK of a signature with extension $\tilde{\sigma} = (\tilde{A}, \tilde{x})$ on the message $(s, \mathsf{usk}, \{s_j\}_{j\in\tilde{\mathcal{I}}}, \{n_j\}_{j\in[1,f]\setminus\tilde{\mathcal{I}}})$ without revealing the signature nor the values s, usk, $\{s_j\}_{j\in\tilde{\mathcal{I}}}$, $\{n_j\}_{j\in[1,f]\setminus\tilde{\mathcal{I}}}$. The only sub-message known by $\mathcal{SP}$ is, obviously, the value s_{i_1}. In the q-SDH case, the user first computes $C_1 = A h^r$ and $C_2 = g^r h^u$, where r and u are randomly chosen in $\mathbb{Z}_p^*$, and next makes the proof

$$\begin{aligned}\textsc{Pok}\ \ (&\tilde{s}, \mathsf{usk}, s_{i_2}, \cdots, s_{i_\ell}, \{n_j\}_{j\in[1,f]\setminus\tilde{\mathcal{I}}}, x, rx, r, s, sx : C_2 = g^r h^s \wedge \\ &1 = C_2^x g^{-rx} h^{-sx} \wedge e(g_1, g_2) e(h_j, g_2)^{s_1} / e(C_1, w) = \\ &e(C_1, g_2)^x e(h, g_2)^{-rx} e(h, w)^{-r} \prod_{j\in\tilde{\mathcal{I}}} e(h_j, g_2)^{-s_j} \prod_{j\in[1,f]\setminus\tilde{\mathcal{I}}} e(h_j, g_2)^{-n_j}).\end{aligned}$$

3.7 Security Issues

We here give some words on the security and efficiency issues that have been described before for multi-service subscription systems.

Appeared in S. K. Katsikas, J. Lopez and M. Soriano (Eds.): TrustBus 2010, volume 6264 of LNCS, pp. 117-128, 2010.

– **Correctness**: this is obvious that a user having subscribe to a service will be able to produce the proof of knowledge underlying the USE protocol.
– **Soundness**: the unforgeability property is verified due to the unforgeability property of the signature scheme with extension. Since the used one is secure (in our case under the q-SDH assumption), it means that an adversary is not able to output a signature on a new message, even with access to the verification public key and to a signing oracle.
– **Anonymity**: this property is verified due to the use of a zero-knowledge proof of knowledge which blinds the subscribed services to the service provider during the USE procedure. There is no way for $\mathcal{SP}$ to make a link between the SUBSCRIBE and the USE procedures other than by breaking the commitment scheme or the zero-knowledge proof of knowledge.
– **Compactness**: It is obvious that our system is compact since the size of the certificate is the one of the signature scheme with extension (in our case (A, x)) which does not depend on the number of subscribed services.

3.8 Profiling vs. Privacy of Scheme 1

With the above system, we have reached a first level of untraceability of the user. In fact, the service provider does not know the identity of the user during the USE protocol, and can not make the link with a SUBSCRIBE or an ADDSUBSCRIBE procedure since the other subscribed services are blinded. With such system, it is clear that the service provider can make some profiling since it knows which set of services a specific user has subscribed. Thus the service provider can make some statistics on the sets of services that are appreciated by users so as to propose new existing services to its customers by using well-chosen advertisements.

In some cases, a user may want to better protect her privacy *w.r.t.* the service provider by not giving her identity when subscribing services. One may think that this goes against profiling but, in the following, we show that the user can be anonymous and sometimes untraceable by the service provider, while permitting some profiling by the $\mathcal{SP}$.

4 Untraceability during Subscription

In this section, we show how $\mathcal{U}$ can be anonymous *w.r.t.* $\mathcal{SP}$ during both the SUBSCRIBE and ADDSUBSCRIBE procedures. For this purpose, we use a variant of the concept of group signatures called Direct Anonymous Attestations.

4.1 Group Signatures and Direct Anonymous Attestations

Concept of group signature. A group signature scheme permits group members to sign messages such that they are anonymous and unlinkable but for a designated authority which is able to revoke the anonymity of a signature. It is possible to design a group signature scheme [13, 1, 3] using a signature schemes with extensions (see Section 3.1) and an encryption scheme [15, 3]. Most

Appeared in S. K. Katsikas, J. Lopez and M. Soriano (Eds.): TrustBus 2010, volume 6264 of LNCS, pp. 117-128, 2010.

of current constructions are based on the same basis. For example, the XSGS scheme [13] uses the above q-SDH based signature scheme with extensions. In this case, the encryption scheme can be the (double) El Gamal encryption [15, 13] or the linear encryption [3].

Concept of DAA. The concept of list signature schemes has been introduced in [10, 9]. It is a variant of group signature schemes which permits, in some cases, to link the signatures from the same user. The same technique has later been used in [4] for Direct Anonymous Attestations (DAA) where the signatures from a group member can be linked if they are related to the same receiver.

The main difference between a group signature and a DAA is the addition, during the signature process, of a value $T = h_{\mathcal{SP}}^{\mathsf{usk}}$ where $h_{\mathcal{SP}}$ is specific to the receiver. Thus, for one group member and one receiver, this value is always the same, and this group member can be traced with T, but two different service providers cannot make any link between two attestations with two different $h_{\mathcal{SP}}$. For this purpose, it should not exist any link between two values $h_{\mathcal{SP}_1}$ and $h_{\mathcal{SP}_2}$ of two different services providers $\mathcal{SP}_1$ and $\mathcal{SP}_2$. This is done by computing e.g. $h_{\mathcal{SP}}$ using a hash function on public values such as name and address of $\mathcal{SP}$.

4.2 Anonymity but Traceability of the User: Scheme 2

In Section 3.4, we have seen that, during the subscription process of Scheme 1, the user has to prove that she has the right to subscribe to some chosen services. For this purpose, U should include the proof of knowledge of usk such that the revealed value upk equals g^{usk}. As shown previously, this also permits us to obtain non repudiation of the user. But as $\mathcal{SP}$ may do the link between upk and the true identity of $\mathcal{U}$, the latter is not anonymous. $\mathcal{U}$ may belong to the group of people who are authorized to access the services provided by $\mathcal{SP}$ but she needs to be anonymous. As we need non-repudiation, the anonymity should be revoked, in case of dispute, in a proven way: we thus need a group signature scheme. But, as one user needs to be recognized by $\mathcal{SP}$ (to ensure the compactness property) during the AddSubscribe, we need a DAA.

Setup. The Setup protocol is different from the one in Section 3.3 since $\mathcal{U}$ needs to be able to produce a DAA. This is done using a GJoin protocol with a "group" manager during the USetup, so that $\mathcal{U}$ now owns a user secret key usk and a group member certificate $\tau = (Z, u)$ such that $Z = (g_1 h_0^{\mathsf{usk}})^{\frac{1}{\gamma+u}}$ (see [13]). The role of the group manager can here be played by $\mathcal{SP}$ in case there is only one service provider (and since the anonymity is also verified *w.r.t.* the group manager) or by any other designated entity with no commercial link with $\mathcal{SP}$. Finally, let $(\theta_1, k_1 = h^{\theta_1})$ and $(\theta_2, k_2 = g^{\theta_2})$ be two pairs of the El Gamal cryptosystem [15, 13].

Subscribe procedure. During the Subscribe process, instead of the proof of knowledge that the committed usk in $C' = h^{s'} h_0^{\mathsf{usk}}$ is related to a revealed and

Appeared in S. K. Katsikas, J. Lopez and M. Soriano (Eds.): TrustBus 2010, volume 6264 of LNCS, pp. 117-128, 2010.

known $\mathsf{upk} = g^{\mathsf{usk}}$, $\mathcal{U}$ needs to prove that the key usk is related to the DAA. This is possible using the subscription process described in Section 3.4 while replacing U by the following one, including a proof that usk is related to a group member certificate $\tau = (Z, u)$ by $Z = (g_1 h_0^{\mathsf{usk}})^{\frac{1}{\gamma+u}}$.

$$\begin{aligned} U = \textsc{Pok}(s', \alpha, \beta, u, u\alpha, \mathsf{usk} : C' = h^{s'} h_0^{\mathsf{usk}} \wedge T = h_{\mathcal{SP}}^{\mathsf{usk}} \wedge \\ T_1 = h^{\alpha} \wedge T_3 = h^{\beta} \wedge T_2/T_4 = k_1^{\alpha}/k_2^{\beta} \wedge \\ e(T_2, g_2)^u e(k_1, w)^{-\alpha} e(k_1, g_2)^{-u\alpha} e(h_0, g_2)^{-\mathsf{usk}} = e(g_1, g_2)/e(T_2, w)), \end{aligned}$$

where $T_1 = h^{\alpha}$, $T_2 = Zk_1^{\alpha}$, $T_3 = h^{\beta}$, $T_4 = Zk_2^{\beta}$, with $\alpha, \beta \in \mathbb{Z}_p^*$. The other steps of the subscription protocol are unchanged and the user finally obtains the signature with extension $\sigma = (A, x)$ on the message $(s, \mathsf{usk}, \{s_j\}_{j\in\mathcal{I}}, \{n_j\}_{j\in[1,f]\setminus\mathcal{I}})$ as before. As the user proves that she belongs to the group of authorized persons, we keep authorization. Moreover, as the DAA can be opened in our case, we also keep non repudiation. Note moreover that $\mathcal{SP}$ can here store on its database the link between the value T and the subscribed services $s_{i_1}, \cdots, s_{i_k}$, with the value $C = h^s h_0^{\mathsf{usk}} \prod_{j\in\mathcal{I}} h_j^{s_j} \prod_{j\in[1,f]\setminus\mathcal{I}} h_j^{n_j}$.

Addition of services. The AddSubscribe protocol is modified as the same way as above, that is replacing the proof that $\mathsf{upk} = g^{\mathsf{usk}}$ by the proof underlying a DAA. As $\mathcal{SP}$ can retrieve the previously subscribed services by using $T = h_{\mathcal{SP}}^{\mathsf{usk}}$ in its database (see above), it can easily make the aggregation of all the services and provide stronger profiling capabilities. The proof V now becomes

$$\begin{aligned} V = \textsc{Pok}(s, \alpha, \beta, u, u\alpha, \mathsf{usk} : C/\prod_{j\in\mathcal{I}} h_j^{-s_j} \prod_{j\in[1,f]\setminus\mathcal{I}} h_j^{-n_j} = h^s h_0^{\mathsf{usk}} \wedge \\ T = h_{\mathcal{SP}}^{\mathsf{usk}} \wedge T_1 = h^{\alpha} \wedge T_3 = h^{\beta} \wedge T_2/T_4 = k_1^{\alpha}/k_2^{\beta} \wedge \\ e(T_2, g_2)^u e(k_1, w)^{-\alpha} e(k_1, g_2)^{-u\alpha} e(h_0, g_2)^{-\mathsf{usk}} = e(g_1, g_2)/e(T_2, w)), \end{aligned}$$

where $T_1 = h^{\alpha}$, $T_2 = Zk_1^{\alpha}$, $T_3 = h^{\beta}$, $T_4 = Zk_2^{\beta}$, with $\alpha, \beta \in \mathbb{Z}_p^*$.

Use Procedure. The Use procedure is the same as the one in Section 3.6 and is not repeated again here. Note that $\mathcal{U}$ does not have to prove the link between usk and her group membership, as for the Subscribe procedure.

Profiling vs. privacy of Scheme 2. With such system, the privacy of the user is more protected than for the Scheme 1 since she is anonymous *w.r.t.* the service provider. Moreover, as we use DAA, the service provider $\mathcal{SP}$ can make the link between the Subscribe and the AddSubscribe procedure regarding services, and consequently the profiling is the same as for Scheme 1.

4.3 The Case of Group Signatures

It is possible to replace a DAA by a group signature. In fact, such solution may seem strange since the group signature provides unlinkability between Subscribe

Appeared in S. K. Katsikas, J. Lopez and M. Soriano (Eds.): TrustBus 2010, volume 6264 of LNCS, pp. 117-128, 2010.

and AddSubscribe while, as $\mathcal{U}$ needs to give her previously obtained services to obtain the aggregation property, we permit $\mathcal{SP}$ to make some link between Subscribe and AddSubscribe. But the use of group signature is interesting since it is possible to study the untraceability of services, that is to prevent the link between Subscribe and AddSubscribe as we will see in the next section.

5 Service Untraceability

In this section, we complete the group signature based solution where $\mathcal{U}$ is anonymous and untraceable and provide a better privacy protection of users. In the scheme 3, we prevent $\mathcal{SP}$ to make any link between a Subscribe and an AddSubscribe procedure. In the scheme 4, $\mathcal{SP}$ is no more able to make any profiling on user preferences since it can not make any link between two subscribed services. All the techniques below are only applicable when using a group signature. In fact, with the above Schemes 1 and 2, $\mathcal{SP}$ can make the link between Subscribe and AddSubscribe by construction and $\mathcal{SP}$ thus necessarily knows which services are subscribed by a unique user.

5.1 Aggregation by the User: Scheme 3

As said in Section 4.3, the Scheme 2 described above when using a group signature provides anonymity and unlinkability of the user *w.r.t.* the service provider. But, as we ask for compactness, the service provider should know the previously obtained services. One solution is to let the user do the aggregation during the AddSubscribe protocol, without revealing the link with the related Subscribe protocol. This way, the user does not have to give to $\mathcal{SP}$ her subscribed services, and thus becomes truly untraceable by $\mathcal{SP}$.

Subscription and use procedures. Using such solution, the Subscribe and the Use procedures of Scheme 2 remain unchanged, except that during the Subscribe, which now includes a group signature, the value $T = h_{\mathcal{SP}}^{\mathsf{usk}}$ is no more used. Thus, the proof of knowledge becomes:

$$U = \textsc{Pok}(s', \alpha, \beta, u, u\alpha, \mathsf{usk} : C' = h^{s'} h_0^{\mathsf{usk}} \wedge T_1 = h^{\alpha} \wedge T_2/T_4 = k_1^{\alpha}/k_2^{\beta} \wedge \\ T_3 = h^{\beta} \wedge e(T_2, g_2)^u e(k_1, w)^{-\alpha} e(k_1, g_2)^{-u\alpha} e(h_0, g_2)^{-\mathsf{usk}} = e(g_1, g_2)/e(T_2, w)),$$

where $T_1 = h^{\alpha}$, $T_2 = Zk_1^{\alpha}$, $T_3 = h^{\beta}$, $T_4 = Zk_2^{\beta}$, with $\alpha, \beta \in \mathbb{Z}_p^*$.

Addition of services. As the user now aggregates the subscribed services by not revealing the previously obtained one, we need to modify the U proof of knowledge. In fact, the user still sends to $\mathcal{SP}$ the commitment on all previously obtained services, that is $C = h^s h_0^{\mathsf{usk}} \prod_{j \in \mathcal{I}} h_j^{s_j} \prod_{j \in [1,f] \setminus \mathcal{I}} h_j^{n_j}$ (see Section 3.5). But, this time, she has to prove that this commitment is well-formed while keeping secret the already subscribed services $s_{i_1}, \cdots, s_{i_k}$. Before that, we remark

Appeared in S. K. Katsikas, J. Lopez and M. Soriano (Eds.): TrustBus 2010, volume 6264 of LNCS, pp. 117-128, 2010.

that the user should not send as it is the above commitment C since this one is known by $\mathcal{SP}$ during the SUBSCRIBE protocol (see Section 3.4). Thus, she has beforehand to modify it. This is done by the user who first chooses at random one $\hat{s} \in \mathbb{Z}_p^*$ and computes $\hat{C} = h^{\hat{s}}C$. Moreover, the user has to prove that she has truly already subscribed the services included into $\hat{C}$ by proving her knowledge of a signature with extension (A, x) on these services.

The user first computes $C_1 = Ah^r$ and $C_2 = g^r h^u$, where r and u are random, and the proof of knowledge V becomes in this case

$$
\begin{aligned}
V = \textsc{Pok}(&s, \hat{s}, x, r, rx, sx, \alpha, \beta, u, u\alpha, \mathsf{usk}, s_{i_1}, \cdots, s_{i_k}, \{n_j\}_{j\in[1,f]\setminus\mathcal{I}} : \\
&\hat{C} = h^s h^{\hat{s}} h_0^{\mathsf{usk}} \prod_{j\in\mathcal{I}} h_j^{s_j} \prod_{j\in[1,f]\setminus\mathcal{I}} h_j^{n_j} \wedge C_2 = g^r h^s \wedge 1 = C_2^x g^{-rx} h^{-sx} \wedge T_1 = h^\alpha \wedge \\
&T_3 = h^\beta \wedge T_2/T_4 = k_1^\alpha / k_2^\beta \wedge e(T_2, g_2)^u e(k_1, w)^{-\alpha} e(k_1, g_2)^{-u\alpha} e(h_0, g_2)^{-\mathsf{usk}} = \\
&e(g_1, g_2)/e(T_2, w) \wedge e(g_1, g_2) e(h_j, g_2)^{s_{i_1}} / e(C_1, w) = \\
&e(C_1, g_2)^x e(h, g_2)^{-rx} e(h, w)^{-r} \prod_{j\in\mathcal{I}} e(h_j, g_2)^{-s_j} \prod_{j\in[1,f]\setminus\mathcal{I}} e(h_j, g_2)^{-n_j}),
\end{aligned}
$$

where $T_1 = h^\alpha$, $T_2 = Zk_1^\alpha$, $T_3 = h^\beta$, $T_4 = Zk_2^\beta$, with $\alpha, \beta \in \mathbb{Z}_p^*$. Next the service provider uses $\hat{C}$, chooses at random $\tilde{s}'$, and computes $\tilde{C} = \hat{C} h^{\tilde{s}'} \prod_{j\in\tilde{\mathcal{I}}\setminus\mathcal{I}} h_j^{s_j - n_j}$. $\mathcal{SP}$ finally computes the final signature with extension $\tilde{\sigma} = (\tilde{A}, \tilde{x})$ on the message $(s, \mathsf{usk}, \{s_j\}_{j\in\tilde{\mathcal{I}}}, \{n_j\}_{j\in[1,f]\setminus\tilde{\mathcal{I}}})$.

Profiling vs. privacy of Scheme 3. On one side, the user privacy is protected since she is anonymous and unlinkable *w.r.t.* $\mathcal{SP}$ all the time. On the other side, $\mathcal{SP}$ is able to make profiling by storing the *subsets* of services users are interested in. In fact, SP can make such profiling for one SUBSCRIBE or one ADDSUBSCRIBE procedure, but not between both such procedures.

5.2 The No-Profiling Case: Scheme 4

The privacy protection can be higher than the previous section, at the cost of a less interesting profiling for $\mathcal{SP}$. The procedures are similar to the previous one, except that SUBSCRIBE and ADDSUBSCRIBE are only used with one single service at a time. This way, the user privacy is completely protected. On the other hand, $\mathcal{SP}$ is no more able to profile users with this solution.

6 Conclusion

We have presented in this paper several schemes which allows users to protect their privacy while permitting them to subscribe to services. In some of our proposals, service providers are moreover able to make some kind of profiling. Note that using current benchmarks on elliptic curve point multiplications and pairing evaluations, our systems can be implemented such that most of the procedures need less than 200 ms to be performed.

Appeared in S. K. Katsikas, J. Lopez and M. Soriano (Eds.): TrustBus 2010, volume 6264 of LNCS, pp. 117-128, 2010.

References

1. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270. Springer, 2000.
2. M. Blanton. Online subscriptions with anonymous access. In *ASIACCS*, pages 217–227. ACM, 2008.
3. D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.
4. E. F. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *ACM Conference on Computer and Communications Security - ACM CCS 2004*, pages 132–145. ACM, 2004.
5. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact E-Cash. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 302–321, 2005.
6. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *SCN 2002*, volume 2576 of *Lecture Notes in Computer Science*, pages 268–289. Springer, 2002.
7. J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Crypto'04*, LNCS, pages 56–72. Springer, 2004.
8. S. Canard, A. Gouget, and E. Hufschmitt. A handy multi-coupon system. In *ACNS 2006*, volume 3989 of *Lecture Notes in Computer Science*, pages 66–81, 2006.
9. S. Canard, B. Schoenmakers, M. Stam, and J. Traoré. List signature schemes. *Discrete Applied Mathematics*, 154(2):189–201, 2006.
10. S. Canard and J. Traoré. List Signature Schemes and Application to Electronic Voting. In *WCC 2003*, pages 81–90, 2003.
11. D. Chaum and T.P. Pedersen. Transferred cash grows in size. In *Eurocrypt'92*, volume 658 of *LNCS*, pages 390–407. Springer, 1992.
12. L. Chen, M. Enzmann, A.-R. Sadeghi, M. Schneider 0002, and M. Steiner. A privacy-protecting coupon system. In *Financial Cryptography 2005*, volume 3570 of *Lecture Notes in Computer Science*, pages 93–108. Springer, 2005.
13. C. Delerablée and D. Pointcheval. Dynamic fully anonymous short group signatures. In *VIETCRYPT 2006*, volume 4341 of *Lecture Notes in Computer Science*, pages 193–210. Springer, 2006.
14. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, 1986.
15. T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO'84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 1984.
16. Mastercard and VIsa. Secure Electronic Transaction (SET), 1996.
17. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*. Springer, 1992.
18. Pino Persiano and Ivan Visconti. A secure and private system for subscription-based remote services. *ACM Trans. Inf. Syst. Secur.*, 6(4):472–500, 2003.
19. C.P. Schnorr. Efficient identification and signatures for smart cards. In *Crypto'89*, volume 435 of *LNCS*, pages 239–252. Springer, 1989.

Appeared in S. K. Katsikas, J. Lopez and M. Soriano (Eds.): TrustBus 2010, volume 6264 of LNCS, pp. 117-128, 2010.