Noname manuscript No.
(will be inserted by the editor)

A Warning on How to Implement Anonymous Credential Protocols into the Information Card Framework

Mikaël Ates · Francesco Buccafurri · Jacques Fayolle · Gianluca Lax

Received: date / Accepted: date

Abstract Unlinkability is a privacy feature supported by those multi-party security protocols allowing anonymous users' credential exchanges among different organizations. Proper signature schemes, based on discrete logarithms, must be used in order to guarantee the above requirements as well as selective disclosure of information. In this paper, we highlight that whenever a concrete architecture based on the above protocols is implemented, some aspects concerning how to manage the association between bases of discrete logarithms and attributes used in attribute certificates should be carefully considered, in order to guarantee that unlinkability really holds. We show that the problem is concrete by testing that the state-of-the-art implementation suffers from the above problem. A general solution is also proposed.

Mikaël Ates
Entr'ouvert,
169 rue du Château,
75014 Paris, France
E-mail: mates@entrouvert.com

Francesco Buccafurri (contact author)
DIMET, University of Reggio Calabria,
via Graziella, Feo di Vito,
89122 Reggio Calabria, Italy
E-mail: bucca@unirc.it

Jacques Fayolle
DIOM, Université de Lyon,
23 rue Michelon,
42023 Saint-Etienne, France
E-mail: jacques.fayolle@univ-st-etienne.fr

Gianluca Lax
DIMET, University of Reggio Calabria,
via Graziella, Feo di Vito,
89122 Reggio Calabria, Italy
E-mail: lax@unirc.it

Keywords Unlinkability · Attribute Certificates · Privacy · U-Prove

1 Introduction

A credential is a powerful means to establish an identity, a role, or an attribute, in order to control accesses to digital services over the Web. There exist many application contexts in which users can be required to show a credential to access a service. For example, a car renting company could deploy an on-line registration process where users are asked to prove the possession of the driving license.

Actually, there does not exist a standard credential exchange architecture, though the main stakes and issues are now well-known [1]. One of the known issues is that credential exchanges can represent a threat on users' privacy. Indeed, as stated by [2], any information that distinguishes one person from another can be used for re-identifying data. For instance, considering the use of the driving license document, this one would induce the user to reveal unnecessary personal information (name, date of birth, and so on).

Thus, in order to protect user's privacy in digital information exchange processes, the credential exchange architecture must permit users to perform a *selective disclosure* of their attributes in credentials, like the proof that an attribute value lies in a given interval (for example, a user could prove to be of age without revealing the date of birth).

Unfortunately, selective disclosure is not enough. Another important threat on privacy arises from the fact that two organizations could exchange the credentials shown by users in order to infer more information about the users. This is related to the issue of *linkabil-*

ity of the user's transaction records hosted by multiple organizations [3]. Due to the grow of attention toward privacy concerns, nowadays *multi-party security* protocols often deal with the above issue.

Two very relevant signature schemes proposed in the literature to guarantee that the signature of attribute certificates is not a factor of linkability are the *CL-Signature signature schema* [4] and the *Secret Key Certificate signature schema* [5]. Both are based on discrete logarithms for representing values of attributes in order to support also selective disclosure. The attribute certificates generated with the above schemes are called *anonymous credentials* and *private credentials*, respectively. Microsoft has recently implemented a multi-party security protocol called *U-Prove* [6,7], based on the scheme [5]. In fact, it represents the state-of-the art implementation of this class of protocols.

It is worth noting that the unlinkability property should be a feature aimed to defend the individual from the privacy threats coming from every party, including those whose position, role and dimension give them a strong control power on users, and, at the same time, a seeming trustworthiness. As a consequence, we cannot exclude in general that even authoritative entities (in principle, also a government organization, for example) could be malicious as far as the privacy issue is concerned.

The design of privacy-preserving cryptographic protocols is a hard task [8], especially when anonymity is addressed, since even cryptographic material can be the place for steganography [9]. Indeed, covert channels can be set-up with X.509 certificates [10] and also within anonymity systems [11]. A credential exchange architecture is thus particularly exposed.

In the literature, there are many results contributing to address the issue of linkability of user transactions. Considering the transactions of certificate issuance and presentation, many factors can lead to linkability, such as the certificate signature, a time correlation, any attribute value being an identifier (for instance a precise date of issuing or an expiry date), any combination of attribute being an identifier. Typically, such issuer-decided data are intrinsically different for each user/holder of a credential. This problem was introduced by Chaum in [12,13]. He also presented a scheme based on blind signature allowing unlinkable certificate issuance and presentation [14,15]. The certified data revealed as a factor of linkability is also widely studied. Revealing information preserving anonymity is usually unified under the wider topic of k-anonymity [16]. For the issue of time correlation the reader may see [17]. The general solution to such a threat is to fix any issuer-decided data that could serve to link specific users. Such data should be posted for anyone to retrieve and redistribute and a trust mechanisms should be applied to make users sure that they are importing and relying on the very same issuer parameters as other users are.

In this paper, by extending a result originally presented in [18], we identify a new factor of linkability that involves issuer-decided data that are common to users. We highlight that the implementation of multi-party security protocols based on discrete logarithms for representing values of attributes should take care of some aspects concerning the association between bases of discrete logarithms and attributes used in attribute certificates. The above association is included into *certificate metadata*. We show that, if the above issue is not correctly handled, then the resulting system would allow adversarial organizations to break the unlinkability of user transactions by establishing a covert channel based on certificate metadata. In particular, such an attack exploits the possibility of choosing the associations between bases of logarithms and the attributes they represent. The issuer could assign specific base/attribute associations to a user or to a set of users (for example, female users), in such a way that a colluding verifier could infer additional information from the transactions.

The relevance of the problem highlighted in the paper is confirmed by the fact that the U-Prove integration into the Identity Metasystem [19] suffers from the above problem. Concerning this issue, we highlight that the problem does not regard the cryptographic protocol itself [20]. Indeed, U-Prove is made up of several interlinked cryptographic protocols and the observed flaw pertains the Microsoft's current test implementation of that technology in the context of the Information Card. The same problem would apply to the implementation of any other anonymous attribute-based credential technology when implemented in this Information Card framework. Before submitting the paper, we contacted the Microsoft's U-Prove Team in order to inform them about our results. We received a feedback both interesting and encouraging. Indeed, they agree that the problem we have identified in the paper really exists in the current Community Technology Preview release of U-Prove. They are experimenting different ways to prevent this issue in the release version.

Another contribution of the paper is to present a practical solution to the above issue based on public certificate metadata retrieved anonymously by the users. Importantly, this solution can be applied to the system U-Prove basically preserving its architecture.

The rest of the paper is organized as follows. In Section 2, some background notions are given and the reference scenario is illustrated. In Section 3, we show how

the certificate metadata can be used in order to implement a covert channel breaking unlinkability. Then, we apply this attack on the U-Prove Technology integrated into the Identity Metasystem V1.0. In Section 4, we present a solution to handle certificate metadata for unlinkable certificates. Finally, in Section 5, we draw our conclusions.

2 Background

In this section, we illustrate the scenario dealt with in our paper and the issue of unlinkability. We describe the signature scheme which the U-Prove architecture is based on and focus on the discrete logarithm used to represent attributes. Finally, we discuss about the certificate metadata.

2.1 The Scenario

The scenario we refer in our paper is illustrated in Fig. 1. We have a user and two organizations. The first organization acts as *(attribute) certificate issuer*, the second as *certificate verifier*. The user plays the roles of *certificate recipient* and *certificate prover*, since she requests first the credential to the issuer and then shows such a credential to the verifier. According to the notation introduced in [21], we denote by the symbol $\nsim$ the unlinkability property. As highlighted in Fig. 1, we want that the transaction of certificate issuance (T1) be unlinkable with the transaction of presentation and proving (T2).

For instance, consider the example described in the introduction. The certificate issuer is a state adminstration which delivers digital driving license. Citizens can retrieve from it a digital certificate corresponding to a driving license. They can store this document and use it whenever they want. Now, a renting company requires the possession of a valid driving license during the registration process. There is the unlinkability of these transactions if the driving license issuance is unlinkable to the registration with the renting company. In other words, it means that two customers which reveal the same information to the renting company are not distinguishable. The benefit on privacy is for instance that if the state administration and renting companies collude, the state administration cannot learn with which renting company citizens have registered. Another benefit is also that if the state administration colludes with one company and reveals it some information about a citizen, the company is not able to associate this extra information with a customer who registers.

2.2 Cryptographic Background

In this section, we provide the cryptographic background necessary to understand the Secret Key Certificate signature schema [20]. For the sake of presentation, some unnecessary details have been omitted.

In a certificate, we have values (for example, "John Smith", "02/04/1980", etc.) of attributes (for example, name, birthdate, etc.). Given an attribute a_i, its value q_i is represented by a discrete logarithm of base g_i. The bases are elements of a modular group of modulus n. Given l attributes, their discrete logarithm representation (hereafter, DL representation) is of the form $\prod_{i=1}^{l} g_i^{q_i} \ mod\ n$. The set of all the bases used for discrete logarithm representations are fixed and included in the public key of the certificate issuer. In particular, the parameters of the Secret Key Certificates signature schema are the following.

Private key: y_0, taken at random in $\mathbb{Z}_q$, with $\mathbb{Z}_q$ the set of integers modulo q.

Public key: $(p,\ q,\ g,\ g_0 = g^{y_0},\ ...,\ g_l = g^{y_l},\ z_0 = g_0^{y_0},\ ...,\ z_l = g_l^{y_0})$ where q is the prime order of a group generated with g and of modulus p, y_0 the private key, y_i with $i \in \{1, ..., l\}$ are taken at random in $\mathbb{Z}_q$.

The certificate issuance is the following sequence.

Let δ and σ_z be two DL representations of the attributes values $q_1,\ ...,\ q_l$:

$$\delta = g_0 g_1^{q_1} ... g_l^{q_l} \ mod\ p \ \ and \ \ \sigma_z = z_0 z_1^{z_1} ... g_l^{z_l} \ mod\ p$$

The certificate issuer chooses at random w in $\mathbb{Z}_q$ and sends to the recipient $\sigma_a = g^w \ mod\ p$ and $\sigma_b = \delta^w \ mod\ p$.

The recipient chooses at random α in $\mathbb{Z}_q$ and β_1, β_2 in $\mathbb{Z}_q^*$, and sends to the issuer

$$\sigma_c = \beta_1 + \mathcal{H}(h, \sigma'_z, \sigma'_a, \sigma'_b)$$

with $\mathcal{H}$ a one-way hash function, $h = \delta^\alpha$, $\sigma'_z = \sigma_z^\alpha$, $\sigma'_a = \sigma_a g_0^{\beta_1} g^{\beta_2}$ and $\sigma'_b = \sigma_b^\alpha \sigma_z'^{\beta_1} h^{\beta_2}$.

The certificate issuer replies with $\sigma_r = \sigma_c y_0 + w \ mod\ q$.

The recipient computes $\sigma'_r = \sigma_r + \beta_2 \ mod\ q$ and $\sigma'_c = \sigma_c - \beta_1 \ mod\ q$ and $\sigma'_z = \sigma_z^\alpha \ mod\ p$.

The signature value is the tuple $(\sigma'_r, \sigma'_c, \sigma'_z)$ and is verified checking that $h \neq 1$ and

$$\sigma'_c = \mathcal{H}(h, \sigma'_z, g^{\sigma'_r} g_0^{-\sigma'_c}, h^{\sigma'_r} \sigma_z'^{-\sigma'_c})$$

2.3 Attribute Certificates

An attribute certificate consists of the attribute values, the signature values, and the certificate metadata. The certificate metadata are used to make the attribute

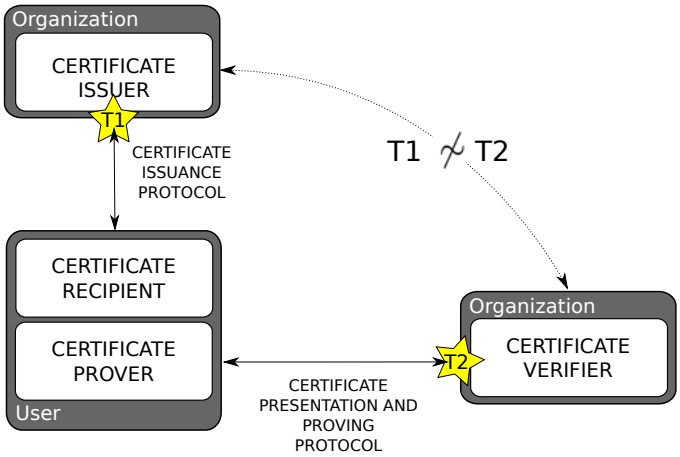

Fig. 1 Unlinkability for attribute certificates.

values and the signature values understandable to the verifier. The metadata include the declaration of the exploited algorithms and the identification of both the signature elements and the signed attributes. In Fig. 2, we show an example of an X.509 certificate [22]. Therein, all metadata are in normal font (excluding the signature value which is underlined) and the values of the certified data are in bold font. In order to guarantee the integrity of metadata, they are also signed (dotted frame).

The metadata used for unlinkable attribute certificates must allow to handle the DL representation of attributes. For this purpose, they must indicate the association between the bases of representation and the types of attribute. For instance, they could indicate that for the attribute *Surname* the fifth base of the bases of representation declared in the issuer parameters has to be used. Thereby, the certificate metadata can be split in three data subsets:

- The semantic of the variable data. For instance, for an attribute indicating the surname of a person, the metadata describe the attribute name, the name space, and any other information necessary to give that attribute a sense.
- The description of the cryptographic material necessary to check the signature: the signature values, the algorithms, and the system parameters.
- The association between the bases of representation and the types of attribute.

3 The Attack: Certificate Metadata as Covert Channel

In this section, we show that if no proper strategy is adopted in the management of certificate metadata, an attack is possible allowing the issuer and the colluding verifier to infer more information about the prover than that actually disclosed by the prover. This attack is based on the possibility of creating a covert channel between the issuer and the verifier exploiting the certificate metadata exchanged between the issuer and the prover and between the prover and the verifier. A covert channel is a communication channel allowing a process to transfer information in a manner that violates the system security policy [23]. We can adapt this definition to our case saying that certificate metadata of an unlinkable attribute certificate are a covert channel if they allow the issuer to transfer to the verifier information about the prover.

In order to be more concrete, consider the following example of covert channel. The malicious issuer can give the provers two different types of certificate metadata, namely t_1 and t_2. The issuer and the colluding verifier agree that the type t_1 will be given only to a particular user u (or to a class of users, for example, male users). For the other users, the type t_2 will be exploited. It is clear that whenever the verifier will receive an attribute certificate whose metadata are of the type t_1, it can deduce that the prover is u (or a male, if the association t_1/male has been adopted) without the prover can detect such an information leak. Conversely, if the metadata type shown by the prover is t_2, the verifier deduces that the prover is not u (or, that the prover is a female). As a consequence, the certificate metadata are a covert channel since they reveal information about the prover, without the prover knowing.

The question we have to address now is: Is it possible to use certificate metadata as a covert channel?

Observe that it is not possible to have metadata that differ for the names (e.g., exploiting the case sensitiveness we could use "Surname" and "surname" for the same attribute) or to use the numeric values of the

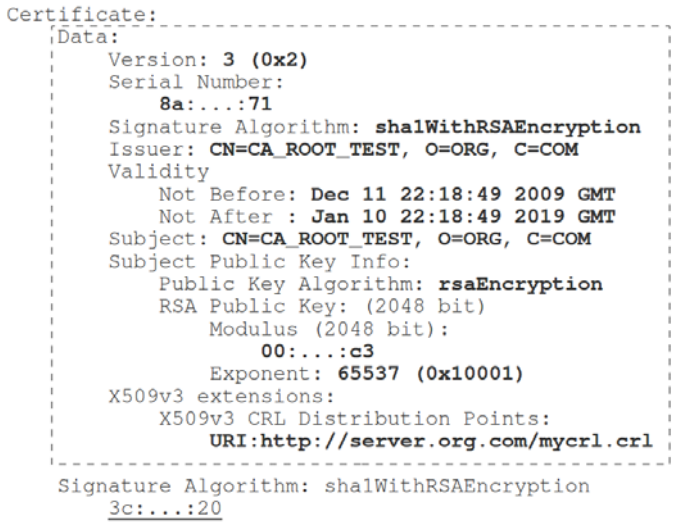

Fig. 2 Metadata in an X.509 certificate.

bases for that purpose. Indeed, the name space has to be case sensitive and common, and the values of the bases are included in the public key of the issuer so that they are fixed and used pervasively.

The possibility we have found to create different types of certificate metadata is exploiting the association between the attributes and the bases used to represent them. This obviously has to be allowed by the implementation of the protocol. Clearly, even though the protocol permits this, an honest issuer should not provide provers with different certificate metadata. Conversely, the malicious issuer and the colluding verifier can set up a covert channel against users which is based on the fact that the issuer gives specific metadata for each user. Said a the number of attributes, the issuer can create $a!$ different associations between attributes and bases, so that it can identify a subset of users with cardinality $a! - 1$, leaving 1 association to deal with the rest of the population.

For example, if the verifier aims to know the gender of provers, it can agree with the issuer to use the two different attributes/metadata associations reported in Table 1, where $g_0, \cdots, g_9$ are the bases of the issuer.

Attribute type	Bases for males	Bases for females
Issuer	g_0	g_3
Date of issuance	g_1	g_7
Date of expiration	g_2	g_0
Surname	g_3	g_8
Firstname	g_4	g_6
Date of birth	g_5	g_1
Gender	g_6	g_2
Address	g_7	g_4
Email	g_8	g_9
Phone	g_9	g_5

Table 1 Order of the bases used as factor of linkability.

3.1 Proof of Concept

In order to demonstrate that the warning detailed in the previous section is concretely relevant, we show that also the U-Prove integration into the Identity Metasystem, specified in [19], is not immune from the attack. The specification has been implemented under the U-Prove Community Technology Preview (CTP) name. These implementations are presented in [24].

The implementation of U-Prove is based on the extensions of Active Directory Federation Services 2.0, Windows Identity Foundation and CardSpace 2.0. The test bench is composed of:

- Two certificate issuers, called *Token Issuers*, one based on Active Directory Federation Services 2.0 CTP and one based on Windows Identity Foundation CTP, hosted by a Windows Server Enterprise 2008 SP2 station.
- A verifier, called *Relying Party*, based on Windows Identity Foundation CTP, hosted by a Windows Server Enterprise 2008 SP2 station.
- Two user environments, one with a Web browser Internet Explorer 7.0 hosted by a Windows Vista SP1 station and one with a Web browser Internet Explorer 8.0 hosted by a Windows Server Enterprise 2008 SP2 station, and both provided with an application for token management, called *Identity Selector*, which is CardSpace 2.0 CTP.

According to the specifications of [19], certificate metadata are called here *issuer parameters*. The issuer

parameters contain the association between bases and attributes. The specifications indicate that during the first round of a token issuance, if the identity selector does not give an identifier of up-to-date issuer parameters, then the issuer provides a response containing the issuer parameters. This response is preceded by a user request containing the user credentials to authenticate to the issuer. The issuer parameters are thus here given when the user is authenticated. Moreover, there is no other means specified in [19] to provide the identity selector with the issuer parameters.

In our test, the user retrieves from the issuer an *Information Card*. Such a document indicates to the identity selector the authentication mechanisms required, applicative endpoints for issuance, and the attributes (called *Claims*) the user can obtain. This document does not contain the issuer parameters. The user adds this document to the identity selector.

By a user action on the relying party, the latter triggers a token request. The user Web browser forwards this request to the identity selector. The identity selector asks the user which issuer to request and the authentication credentials. With both the issuers deployed, we have observed that whenever the identity selector performs the first token request without indicating an identifier of up-to-date issuer parameters, the issuer replies with a message response containing the tokens and the issuer parameters.

Then, the tests confirm that the configuration options offered by CardSpace 2.0 CTP do not permit to add issuer parameters. As a consequence, before the first token request, there is no way for the user to obtain the issuer parameters. The user is then provided with the issuer parameters at the token issuance and these issuer parameters are the ones used by the identity selector to verify the token validity and to lead the proofs.

Finally, we have tested the system with multiple issuer parameters and verified that correct orders for associations between discrete logarithm bases and attribute types are required for successful user proofs.

We conclude that this method is the unique way to provide users with issuer parameters in the U-Prove implementation. Since the issuer parameters are given when the user is authenticated and the user has no means to check the uniqueness of the issuer parameters in all the realm, the association of bases and attributes can be user-specific and the attack can be carried out.

4 The Solution

In this section, we present a possible solution to the attack described in the previous sections. First, in Section 4.1, we illustrate how the solution strategy proceeds and we show its effectiveness. Then, in Section 4.2, we implement this solution in the U-Prove architecture.

4.1 Description

We have seen that the general solution to the problem of issuer-decided data is to fix their value. This is possible also in our case, where the covert channel underlying the attack is implemented by changing the association between bases and attributes in the attribute certificates. In order to make users sure that they are importing and relying on the very same issuer parameters as other users are, a possibility is to exploit a public key infrastructure certification where a certification authority (CA) publishes, by an issuer setup certificate, the *legal* association and ensures that the association is unique for all the users of a given issuer. For instance, the user, before the certificate proving, might download the certificate metadata for the considered issuer from CA in order to check whether the issuer is adopting the legal association. This solution clearly works, but it is in practice little feasible, since it results in a strictly hierarchical architecture strongly limiting the pervasiveness of the system (for example, think of the management of join and leave of issuers). This is in fact coherent with the choice done in the U-Prove architecture, which does not adopt any rigid hierarchy on top of the issuers. Moreover, our attack is based on the assumption that even authoritative entities (in principle, also a government organization, for example) could be malicious as far as the privacy issue is concerned. In fact the unlinkability property should be a feature aimed to defend the individual from the privacy threats coming from every party, including those whose position and dimension give them a strong control power on users, and, at the same time, a seeming trustworthiness. Under this assumption, it is difficult to identify in the real case which entity could play the role of CA.

Due to the above considerations, we propose a solution preserving the architecture of U-Prove and relying only on the autonomous ability of the user to check the trustworthiness of the issuer.

The solution, which we call *two-phase-issuance (2PI)*, consists in dividing the issuance step into two distinct phases:

1. The user retrieves from the issuer the certificate metadata anonymously.

2. The user retrieves from the issuer the signature values on a set of attribute values, without revealing any information about certificate metadata previously obtained.

The solution expects that time-correlation attacks are not applicable, but this is typical in the context of unlinkability. Indeed, if this is not the case, the time correlation between issuance and proving steps would reveal the user identity.

Observe that, the above protocol can be obtained by using the features of U-Prove, as we describe in Section 4.2.

We next show that the above solution works, in the sense that the probability that unlinkability is broken by means of a malicious behavior of issuers and verifiers is the same as it happens whenever the two parties guess (with no a-priori knowledge) this linking information. We note that the unlinkability is broken if a pair issuer-verifier is able to distinguish a subset of users.

Consider a pair issuer-verifier, say I and V. Let denote by U the set of all users. Let u be a subset of users characterized by some values (or ranges) of the attributes. Obviously, if V tries to distinguish the subset u just by guessing, the success probability is $\frac{|u|}{|U|}$.

Consider now the case that I and V agree on a particular association bases-attributes in order to identify u. For example, they want to distinguish male and female users. The issuer generates two associations, say A_M and A_F, and the expected goal is to assign A_M to male users and A_F to female users. Thanks to *2PI*, there is no way to deterministically know if the user retrieving anonymously certificate metadata (i.e., issuer parameter in U-Prove terminology) is male or female. As a consequence, the only possibility is to guess this feature. Clearly, if the guessing succeeds, then the user will not be able to detect the malicious behavior since the certificate metadata obtained in the second phase of the protocol *2PI* coincides to those obtained before. In the general case, the only possibility for I to implement a cover channel allowing V to link the subset u is to guess that the user requiring certificate metadata in phase 1 of *2PI* is belonging to u. The attack succeeds only if this happens, thus with probability $\frac{|u|}{|U|}$.

The solution thus fully preserves unlinkability, since there is no higher probability for V to infer linking information thanks to the colluding issuer I w.r.t. the case that V cannot rely on the cooperation of I.

4.2 Mechanisms to Handle the Certificate Metadata: Application to U-Prove

In this section, we describe how to implement the solution in the U-Prove architecture.

From the previous analysis we have deduced that the certificate metadata must be served to users independently from the certificate issuance. We can thus design one document per issuer gathering all certificate metadata. The content of such a document is illustrated in Fig. 3. The certificate metadata must be signed by the issuer to ensure the users and verifiers of their integrity. Notice that differently from other kind of certificates, like X.509 certificates, the certificate metadata and the attribute values are not signed together.

The issuer publishes the certificate metadata in a public repository accessible anonymously. This document is obtained by a third protocol dedicated to this purpose. The users retrieve this document anonymously and use it to verify a certificate, using also the signature values obtained and the attribute values expected. The verifiers also retrieve this document. When they validate a proof, they associate quantities with bases. Then, they deduce the corresponding attribute from the certificate metadata.

These mechanisms are illustrated in Fig. 4. The transactions numbered 2, 3 and 4 are unlinkable.

Observe that the certificate metadata could be formatted using any standard of document formatting. For instance, with XML, the U-Prove issuer parameters name space may be used. The X.509 version 3 certificate extension field capabilities may also be used [25]. The parameters would be added in extension fields of an X.509 v.3 certificate of a certification authority for instance. This certificate would have to be distributed with the constraints previously defined. The *Subject Public Key Info* field of an X.509 public key certificate contains the public key description. However, the algorithms employed for the Secret Key Certificates are not identified in the standard specification. Thus, the public key parameters should be included in a specific extension field. Moreover, extension fields to describe the attributes and the associations with the logarithmic bases must be defined. As a consequence, in the process of standardization, the definition of a new X.509 extension profile would be necessary.

These guidelines can be applied to the U-Prove architecture. The certificate metadata should substitute the issuer parameters. However, the identity selector should be able to retrieve the issuer parameters anonymously in a first separate step. Then, it is necessary

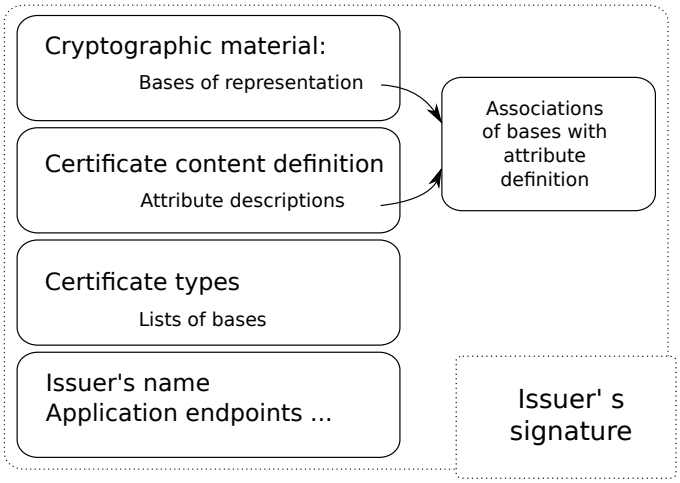

Fig. 3 Example of certificate metadata.

to remove from the protocol all the data related to the issuer parameters during the issuance and presentation protocol. In order to implement a mechanism devoted to verify that the identity selectors are provided with up-to-data issuer parameters, the identity selectors should perform the checking anonymously.

5 Conclusion

In this paper, we have highlighted a possible risk of vulnerability arising from the implementation of multi-party security protocols based on discrete logarithms for representing attributes. In particular, we have shown that if the issuer is free to manage maliciously the association between bases of discrete logarithms and attributes used in attribute certificates, then a covert-channel-based attack is possible allowing colluding issuers and verifiers to break unlinkability enforced by the protocol. We have identified the problem by defining how the covert channel can be implemented and checked that this problem is not only an abstract hypothesis, but a concrete issue. We have reached this conclusion by checking that the most important existing system aimed to provide unlinkable multi-party credential exchange, which is U-Prove, allows malicious organizations to implement the above covert channel, thus potentially breaking unlinkability. The paper addresses also the issue of the prevention of the above risk, by proposing a solution easily applicable also to the concrete architecture of U-Prove. Even though the paper includes some implementation issues which we have applied to the case of U-Prove in order to incorporate in it our solution, it could be interesting to implement a complete system prototype extending U-Prove in the direction we have identified. This is a matter of our future work.

Acknowledgments

We are very grateful to Microsoft Corporation for the interest shown in our work. This work was partially funded by the French Ministry of Economy and Industry and by the Italian Ministry of Research through the PRIN Project EASE (Entity Aware Search Engines).

References

1. Bhargav-Spantzel, A., Camenisch, J., Gross, T., Sommer, D.: User centricity: a taxonomy and open issues. In: DIM '06: Proceedings of the second ACM workshop on Digital identity management, New York, NY, USA, ACM (2006) 1–10
2. Narayanan, A., Shmatikov, V.: Myths and fallacies of personally identifiable information. Commun. ACM **53**(6) (2010) 24–26
3. Pfitzmann, A., Kohntopp, M.: Anonymity, unobservability, and pseudonymity - a proposal for terminology. In: Lecture Notes in Computer Science: Designing Privacy Enhancing Technologies. Volume 2009., Springer Berlin / Heidelberg (2001) 1–9
4. Camenisch, J., Lysyanskaya, A.: A signature scheme with efficient protocols. In: International Conference on Security in Communication Networks - Lecture Notes in Computer Science. Volume 2576. (2002) 268–289
5. Brands, S.A.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press, Cambridge, MA, USA (2000)
6. U-Prove: Microsoft Corporation Technology (2010) `https://connect.microsoft.com/content/content.aspx?contentid=12505\&siteid=642` [Online; accessed 1-September-2010].
7. Brands, S.: U-prove technology overview v1.0. Technical report (2010)
8. Balopoulos, T., Gritzalis, S., Katsikas, S.: Specifying and implementing privacy-preserving cryptographic protocols. International Journal of Information Security **7** (2008) 395–420 10.1007/s10207-008-0057-y.
9. Ahn, L.V.: Public-key steganography. In: In: Advances in Cryptology Proceedings of Eurocrypt 04, Springer-Verlag (2004) 323–341

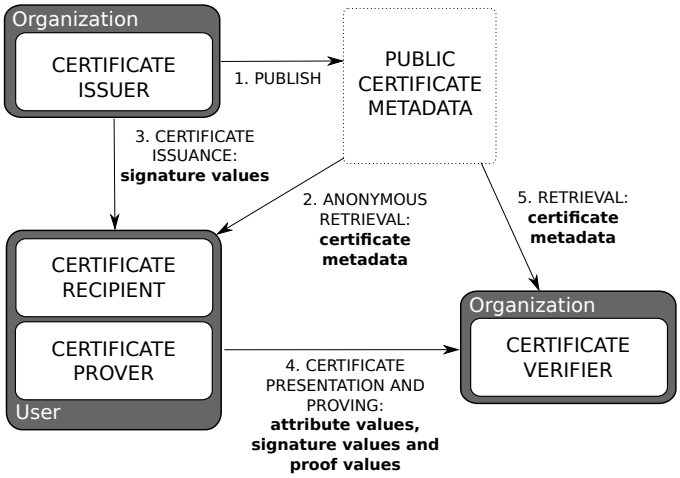

Fig. 4 Implementation of the *two-phase-issuance (2PI)*.

10. Scott, C.: Network covert channels: Review of current state and analysis of viability of the use of x.509 certificates for covert communications. Technical report (2008)
11. Murdoch, S.J.: Covert channel vulnerabilities in anonymity systems. Technical report (2007)
12. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2) (1981) 84–90
13. Chaum, D.: Security without identification: transaction systems to make big brother obsolete. Commun. ACM **28**(10) (1985) 1030–1044
14. Chaum, D.: Blind signatures for untraceable payments. In: International Cryptology Conference on Advances in Cryptology. (1983) 199–203
15. Chaum, D., Fiat, A., Naor, M.: Untraceable electronic cash. In: International Cryptology Conference on Advances in Cryptology, London, UK, Springer-Verlag (1990) 319–327
16. Ciriani, V., De Capitani di Vimercati, S., Foresti, S., Samarati, P.: Theory of privacy and anonymity. In Atallah, M., Blanton, M., eds.: Algorithms and Theory of Computation Handbook (2nd edition). CRC Press (2009)
17. Pashalidis, A., Mitchell, C.: Limits to anonymity when using credentials. In Christianson, B., Crispo, B., Malcolm, J., Roe, M., eds.: Security Protocols. Volume 3957 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2006) 4–12 10.1007/11861386_2.
18. Ates, M.: Digital Identities : User Centric and Privacy-Respectful Cross-Organizational Identity Management. PhD thesis, Université de Lyon - SATIN Team DIOM Laboratory Telecom Saint-Etienne University of Saint-Etienne (2009)
19. Paquin, C.: U-prove technology integration into the identity metasystem v1.0. Technical report (2010)
20. Brands, S., Paquin, C.: U-prove cryptographic specification v1.0. Technical report (2010)
21. Steinbrecher, S., Kpsell, S.: Modelling unlinkability. Lecture Notes in Computer Science **2760** (2003) 32–47
22. Housley, R., Ford, W., W.Polk, Solo, D.: Internet X509 Public Key Infrastructure Certificate and CRL Profile. In: IETF RFC 2459. (1999)
23. US-DoD: Trusted Computer System Evaluation. U.S. Department of Defense. The Orange Book. Publication DoD 5200.28-STD (1984)
24. Paquin, C., Thompson, G.: U-prove ctp white paper. Technical report (2010)
25. Housley, R.: Internet X. 509 Public Key Infrastructure Certificate and Certification Revocation List (CRL) Profile. RFC 3280 (2002)