Untraceable Secret Credentials: Trust Establishment with Privacy[1]

Laurent Bussard, Yves Roudier, and Refik Molva
Institut Eurécom, Corporate Communications
2229, route des Crêtes BP 193
06904 Sophia Antipolis (France)
{bussard, roudier, molva}@eurecom.fr

Abstract

There is generally no a priori trust relationship among entities interacting in pervasive computing environments which makes it necessary to establish trust from scratch. This task becomes extremely challenging when it is simultaneously necessary to protect the privacy of the actors involved. This paper shows how trust can be based on previous interactions yet remain unlinkable to any previous event or any specific entity. A solution based on group blind signatures is proposed that relies on credentials both secret, meaning that they contain an encrypted description of previous interactions, and untraceable, meaning that they cannot be recognized when presented to their issuer.

1. Introduction

The large scale deployment of pervasive computing applications heavily depends on the assurance of essential security properties for users and service providers. In addition to security exposures due to the underlying mobile and wireless communications, pervasive computing applications bring up new security issues. In this paper we tackle two major security problems of pervasive computing. First, such environments lack organization, hence lack *a priori* trust among parties. It is thus impossible to rely on a public key infrastructure and identity based authentication is meaningless [11]. In other words, trust relationships have to be started from scratch. Second, privacy is a major concern of pervasive computing. It is important to ensure that intrusive technology cannot spy users by tracing them and by recording their acts.

The solution to both issues proposed in this paper is to authenticate entities based on their interaction history. We foresee such history as being made of credentials proving that some interaction indeed occurred. For instance, an actor can prove that he was previously certified as a reliable partner by the entity he is interacting with again. After any interaction, a credential is provided in order to subsequently assert what happened in a previous relationship. When two entities interact, they can exchange one or more credentials that enable trust establishment. To ensure that a credential holder will show negative as well as positive statements, we propose to encrypt credentials so that only the issuer and some trusted relatives can open it.

Ensuring privacy in this context means that credential issuers cannot trace users by means of the credentials they delivered them. More precisely, a credential has to be created or modified in a way that forbids the issuer to recognize the credential when it is presented. It is possible to use blind signature mechanisms to ensure that the message and signature cannot be recognized. And, it is necessary to have a way to verify that the secret attribute is the encryption of one element of a public set of cleartexts. Otherwise, if the holder could embed any encrypted attribute, he could attach a unique identifier to each credential in order to trace holders. The technique we propose is to prove that the secret is the encryption of an element of a public set of values. This makes it possible for a credential holder to prove his history of interactions to the issuer or to one of the issuer's relatives without being linkable to a previous event (untraceability) and without revealing his identity (anonymity).

The rest of this paper is organized as follows: Section 2 presents the requirements for history-based trust establishment with privacy and discusses related works. Section 3 describes proofs of knowledge, signatures based on a proof of knowledge, and an existing group blind signature scheme which serves as the basis for our technique. Finally, Section 4 details how this signature scheme can be modified to enable trust establishment when privacy has to be ensured.

[1] Institut Eurécom's research is partially supported by its members: Bouygues Télécom, Cegetel, France Télécom, Hasler Foundation, Hitachi, STMicroelectronics, Swisscom, Texas Instruments, and Thales

2. Problem Statement

Our scenario throughout the paper will be as follows: Alice (A) meets some entity (B), she works with him and gives him a credential to describe their relationship. Subsequently, B comes back and shows his credential to A.

2.1. Expected Features

In order to achieve attribute secrecy while protecting B's privacy, the credential scheme has to fulfill the following requirements:

- The issuer A must not be able to recognize the credential that has been unblinded and thus when B comes back, A does not know that she is talking to the same entity.
- The credential value is secret but has a restricted number of values:
  - B can verify that the credential's secret attribute is part of a public set of values, e.g. {*very poor, poor, fair, good, excellent*} or {0, 1, 2}. The cardinality gives to B a good estimate of the absence of risk that A may use secret values as covert channels for tracing B.
  - The holder B cannot decrypt the attribute of a credential. In other words, B cannot know whether he was described as *good* or *poor*.
  - Probabilistic encryption ensures that B has no way to check whether two credentials embed the same secret attribute value.
- The issuer A and her trusted relatives (e.g. A_2) can retrieve secrets embedded in credentials signed by A.

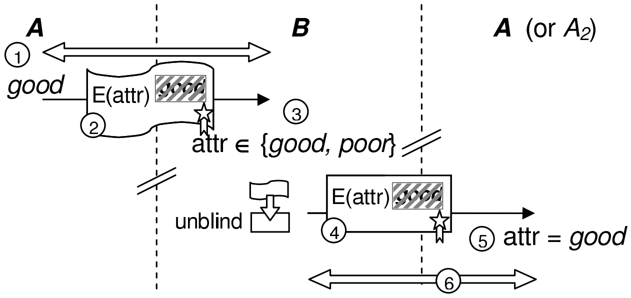

Figure 1. Unlinkable secret credential

Figure 1 presents the different steps for establishing trust: 1) after an interaction, A decides to tag B as *good*. 2) A credential containing this tag is provided to B. 3) The credential attribute is encrypted so that B cannot know its value but B can verify that the value is in a restricted set of possible values, e.g. *bad* or *good*. 4) The credential is unblinded, i.e. modified so that it cannot be traced by A. 5) During a new interaction, B shows his unblinded credential to A or to a relative of A. The possibility that the credential contain a positive feedback, which B cannot evaluate, represents an incentive for B to show his credential to A. 6) The new interaction depends on history but cannot be linked to any specific event like step 1).

2.2. Related Work

Attribute certificates (X.509, SPKI) generally do not protect the privacy of holders that can be identified and traced each time they show a certificate. Privacy-preserving (i.e. anonymous and/or untraceable) attribute certificates are proposed in some works that rely on blind signatures [3], signatures of knowledge [1], or pseudonyms [2].

Establishing and verifying trust relationships is a common problem of *ad hoc* networks. Mechanisms to deal with trust are mainly based on rewards/penalties [6] or on reputation [8]. Privacy is not taken into account in those approaches.

In this paper we show how it is possible to establish trust when privacy is a major concern.

3. Required mechanisms

This section presents group blind signature mechanisms that are used for defining untraceable secret credentials in Section 4.

3.1. Overview

A group signature scheme allows group members to sign messages on behalf of the group. Signatures can be verified with a group public key but do not reveal the signer's identity. Moreover it is not possible to find out whether two signatures have been done by the same member. Finally, only the group manager can *open* signatures, i.e. reveal the identity of the signer.

A blind signature is a protocol in which a signer signs some message m without seeing this message. It was introduced by Chaum [3] to ensure untraceability of electronic cash.

A group blind signature is a protocol in which a group member blindly signs a message. Only the manager can know who signed the message and nobody can recognize the unblinded message.

All existing group blind signature schemes [10] and [9] are based on the group signature schemes proposed by Camenisch in [4]. The conclusion of [5] gives a quick sketch of two other possible approaches. In this paper, we only describe the first blind group signature scheme [10] and modify it so that it fulfills requirements of untraceable secret credentials. However, it seems possible to modify other schemes as well. The remaining of this section briefly presents the first group signature schemes that relies on signatures based on a proof of knowledge.

3.2. Interactive Proof of Knowledge

A *proof of knowledge* (PK) allows an entity to prove the knowledge of some secret without revealing this secret. For instance, a prover P claim to know the double discrete logarithm of y to the bases g and a. A verifier V tests if P indeed knows x. This is denoted $\mathrm{PK}[\alpha \mid y = g^{(a^{\alpha})}]$ where $n = pq$, p and q are two large primes, G is a cyclic group of order n generated by some $g \in G$, and finally $a \in \mathcal{Z}_n^*$.

P sends a witness to V: $w = g^{(a^r)} \bmod n$ where $r \in_R \mathcal{Z}_n$ and V returns a random challenge bit $c \in_R \{0, 1\}$. Finally P sends a response $s = r$ (if $c = 0$) or $s = r - x$ (if $c = 1$). The verifier checks that

$$\begin{array}{lll} c = 0 : & w \overset{?}{=} g^{(a^s)} & = g^{(a^r)} \\ c = 1 : & w \overset{?}{=} y^{(a^s)} & = \left(g^{(a^x)}\right)^{(a^s)} = g^{(a^{x+s})} = g^{(a^r)} \end{array}$$

To be sure that P knows the double discrete logarithm of y (i.e. with a low enough probability of successful attack), this protocol is run l times where l is a security parameter.

3.3. Signature based on a Proof of Knowledge

It is possible to obtain a non-interactive version of the previous scheme. Moreover, when the challenge (i.e. set of challenge bits) depends on a message, it becomes a *signature based on a proof of knowledge* (SPK) or signature of knowledge. For instance the signature of knowledge of a double discrete logarithm of y to the bases g and a, on message m, is denoted $\mathrm{SPK}[\alpha \mid y = g^{(a^{\alpha})}](m)$.

The signature is an $l + 1$ tuple $(c, s_1, \dots, s_l)$ satisfying the equation $c = \mathcal{H}(m \parallel y \parallel g \parallel a \parallel P_1 \parallel \dots \parallel P_l)$ where $P_i = g^{(a^{s_i})}$ (if $c[i] = 0$) or $P_i = y^{(a^{s_i})}$ (if $c[i] = 1$) where $c[i]$ is the i^{th} bit of c. The signature is computed as following:

1. For $1 \leq i \leq l$, generate random r_i.
2. For $1 \leq i \leq l$, set $P_i = g^{(a^{r_i})}$
3. Compute $c = \mathcal{H}(m \parallel y \parallel g \parallel a \parallel P_1 \parallel \dots \parallel P_l)$.
4. For $1 \leq i \leq l$, set $s_i = \begin{cases} r_i & \text{if } c[i] = 0 \\ r_i - x \bmod n & \text{if } c[i] = 1 \end{cases}$

Such a signature means that '*someone knowing the double discrete log of y signed message m*'. Similar signatures can be based on other proofs of knowledge: discrete logarithm, e^{th} root of discrete log, representation, equality of discrete logarithms, etc.

3.4. Camenischs Group Signature

In [4], a group signature requires two signatures of knowledge: one to prove that the signer knows a secret x and another one to prove that his secret is certified by the group manager. The scheme relies on the hardness of computing discrete logarithm, double discrete logarithm and eth root of the discrete logarithm.

The public key of a group is $(n, e, G, g, a, \dots)$ where e is chosen so that $\gcd(e, \phi(n)) = 1$ and $d \cdot e = 1 \bmod \phi(n)$. The private key of the manager is (p, q, d). When $A\ joins$ the group, i.e. becomes a member, she uses her secret x to compute a membership key (y, z) where $y = a^x \bmod n$ and $z = g^y$. A sends (y, z) to the group manager, proves that she knows x and receives a group certificate $(y+1)^d \bmod n$ corresponding to her secret x. In order to sign a message m, A chooses $r \in_R \mathcal{Z}_n$ and computes $\tilde{g} = g^r$, $\tilde{z} = \tilde{g}^y\ (= z^r)$, and two signatures:

$$\begin{array}{l} V_1 = \mathrm{SPK}[\alpha \mid \tilde{z} = \tilde{g}^{(a^{\alpha})}](m) \\ V_2 = \mathrm{SPK}[\beta \mid \tilde{z}\tilde{g} = \tilde{g}^{(\beta^{e})}](m) \end{array}$$

V_1 is a signature of knowledge of a double discrete logarithm that can only be computed when knowing the secret x. Similarly, V_2 is a signature of knowledge of an eth root of the discrete logarithm that can be computed using the certificate $(y + 1)^d$. The group signature of message m is $(\tilde{g}, \tilde{z}, V_1, V_2)$.

The verifier checks that V_1 and V_2 are valid signatures of m and combines both signatures: $\tilde{g}^{(\beta^e)} = \tilde{z}\tilde{g} = \tilde{g}^{a^{\alpha}+1}$. Thus he verifies that $\beta = (a^{\alpha} + 1)^d$ and knows that someone holding a certified secret signed m. However, the verifier cannot know which secret x was used. In other words the identity of the signer is preserved: it is a group signature.

3.5. Group Blind Signature

In the remaining of this paper, we use the group blind signature scheme proposed in [10], which is a blind version of the group signature scheme of Camenisch presented in the previous section. A more efficient group blind signature scheme is described in [9]. It is based on another group signature scheme described in [4] and could potentially be used for our purpose but its security remains uncertain.

The following notations have been used: the public key of group G is $K_{P_G} = \{n, e, G, g, a, \dots\}$, the private key of group member A is $K_{S_A} = \{x, (a^x + 1)^d\}$.

4. Untraceable Signed Secret

This Section shows how the group blind signature scheme can be used to provide an untraceable signed secret which constitutes a basic building block of privacy-preserving trust establishment.

4.1. Principle

Untraceability is guaranteed by the blind signature mechanism. However, it is necessary to associate some attribute value to this signature. We propose to assign to each signer a set of private keys, e.g. $\{K_{S_0}, K_{S_1}\}$. The signer chooses the key according to the attribute value. For instance, a random number signed with key K_{S_0} has a different meaning than any value signed with key K_{S_1}. To enable attribute secrecy, group signature scheme is required: when all private keys are part of a same group, the verifier cannot know which key was chosen and thus cannot discover the attribute value.

A new group is created for each entity that signs secrets. The group key becomes his public key and the same signer uses different private keys according to the value of the attribute that has to remain secret (see right part of Figure 2). In other words, the blind signature ensures the holder untraceability and the group signature yields the secrecy of attributes.

For instance, a signer which can use attribute values from the set $\{$*0:poor, 1:fair, 2:good*$\}$, will have a group public key K_{P_G} and three private keys $K_{S_{G,0}}$, $K_{S_{G,1}}$, and $K_{S_{G,2}}$. When the signer wants to encrypt the value *good*, he blindly signs with the corresponding private key $K_{S_{G,2}}$. Anybody can verify that the unblinded message has been signed with a private key corresponding to the public key K_{P_G}. However, nobody can know which key was used. When the unblinded message is subsequently shown to the signer, he cannot trace the holder but can *open* the signature to know which key was used and can thus retrieve the secret value.

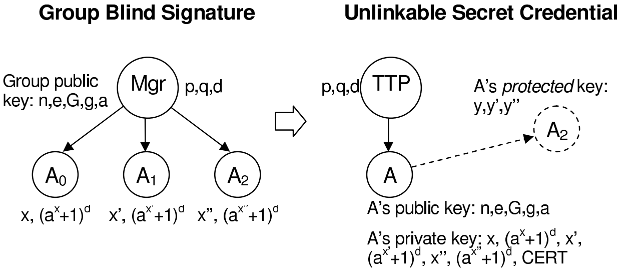
Group Blind Signature
Unlinkable Secret Credential
Group public key: n,e,G,g,a
Mgr
p,q,d
A0
A1
A2
x, (a^x+1)^d
x', (a^x'+1)^d
x", (a^x"+1)^d
p,q,d
TTP
A's protected key: y,y',y"
A2
A
A's public key: n,e,G,g,a
A's private key: x, (a^x+1)^d, x', (a^x'+1)^d, x", (a^x"+1)^d, CERT

Figure 2. Scheme modification

4.2. Restricting Possible Values of a Signed Secret

Unfortunately, the chosen group signature scheme allows new members to join the group without modifying the group public key. In other words, it is not possible to know how many private keys exist for a given group public key. In the context of this paper, it means that the cardinality of the set of values that can be hidden in the secret attribute cannot be deduced from the public key. To solve this problem it is necessary that the 'group' manager role be assumed by a trusted third party (TTP). This TTP provides a set of private keys to each signer and certifies each public key with the number of related private keys that have been created. In this manner, it is possible to ensure that the set of keys is fixed and that the secret attributes can only be the encryption of an element of a public set.

4.3. Protected Keys

Another expected feature is the possibility to let some relative open signatures. Table 1 shows a three stage key scheme: the *private key* is used to sign a credential with a secret attribute and is kept secret by signers. The *protected key* enables the signature verification and access to the secret attribute value (*open*) and is only distributed to trusted partners of the signer. The *public key* enables the verification of the signature and the set of possible values without revealing the secret value. Intermediate keys are said to be *protected*: this terminology was chosen by analogy with object oriented programming languages where access to methods can be defined as public, protected, or private. Each employee of a company could be allowed to open credentials signed by coworkers in order to establish trust relationships in a distributed way.

5. Trust Establishment Protocol

According to the initial scenario and previous section, the following actors are defined: A is an issuer (e.g. corporation, group) that provides credentials to entities that interact with her. B is a holder that collects credentials from different entities in order to build a history. *TTP* is a trusted third party that issued A's keys. A_2 is a partner of A.

5.1. Protocol Description

Before any interaction, A starts i times a *join* protocol with the TTP, i being the number of different values that can be attached to a credential, e.g. with $\{$*0:poor, 1:fair, 2:good*$\}$, $i = 3$. As shown in the right part of Figure 2, A knows i secrets $x, x', \ldots, x^{(i)}$ and receives i membership certificates $(y+1)^d, (y'+1)^d, \ldots, (y^{(i)}+1)^d$. A also receives a public key certificate CERT $= \text{SIGN}_{TTP}(K_{P_A}, i)$, which guarantees a set of possible values. Private, protected, and public keys are distributed according to Table 1.

When A wants to provide a credential to B, the following exchange occurs: B chooses a random message m, A blindly signs this message with the private key corresponding to the chosen attribute value. B verifies with the public key of A that the signature is correct. The certificate *CERT*

Knowledge / Capability	**TTP**	**A**	**A$_2$**	**B**
A's public key $\{n, e, G, g, a\}$, CERT	☒	☒	☒	☒
A's protected key $\{y, y', \ldots\}$	☒	☒	☒	□
A's private key $\{x, (y+1)^d, x', \ldots\}$	□	☒	□	□
TTP's secret on A $\{p, q, d\}$	☒	□	□	□
Verify signature of A and attr $\in$ set	☒	☒	☒	☒
Retrieve value of secret attribute	☒	☒	☒	□
Sign credential as A	□	☒	□	□
Define set of attribute values	☒	□	□	□

Table 1. Distribution of secrets among actors

is public and defines the set of possible values in the secret attribute.

When B gets in touch with A or (A_2), he shows an unblinded version of the credential and A *opens* the signature to retrieve the secret attribute. The only information available to A is that she is interacting with an entity that she had previously tagged as *good*.

5.2. Security Evaluation

Untraceable secret credentials are based on a group blind signature scheme that has been shown secure [10, 4]. These group blind signatures are however used in a particular manner in this paper: signers receive multiple private keys and protected keys have to be defined and distributed.

Because A acts as multiple group members and knows related secrets and certificates, it is mandatory that the group signature scheme be resistant to coalition attacks. The initial *join* protocol of [4] has to be replaced by a more secure one. This modification is taken into account in the used group blind signature scheme [10].

Distributing *protected keys* (y, y', etc.) to partners (e.g. A_2) does not weaken the scheme. Partners as well as manager cannot impersonate group members and partners cannot enable covert channels (new members) because they do not have access to TTP's secrets.

6. Conclusion and Future Work

This paper presented a technique of untraceable secret credentials enabling privacy-preserving history-based trust relationships. Secrecy ensures that positive as well as negative statements can be used in the behavior description attached to an entity, which is infeasible with cleartext negative statements that holders can simply choose not to present. Blind group signature scheme is shown to be a possible mechanism for implementing the untraceability of such a history based credential holder. This represents a first step towards full-featured untraceable secret credentials, yet some issues have to be solved:

- Non-transferability is weakly ensured. It can be assumed that the secrecy of credentials makes it impossible to trade them. Moreover, it is possible to keep a trace of used credentials and to reject one that was previously used. However, a message that is blindly signed should be linked to a valuable secret of the holder.
- A trusted third party is required to certify that the number of possible values of secret attributes is restricted. Indeed, the group's public key does not depend on the group size and thus knowing the public key is not sufficient to know the number of group members, i.e. the set of possible attribute values. Whether it is possible to render blind a group signature scheme with a public key depending on the number of members, and thus to suppress the need for a TTP, is still an open issue.

We are trying to solve these limitations and are simultaneously working on a higher-level privacy-preserving trust and history management architecture.

References

[1] S. Brands. *A technical Overview of Digital Credentials*. Research Report, February 2002.

[2] J. Camenisch and A. Lysyanskaya, *An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation*, LNCS 2045, 2001.

[3] D. Chaum and R.L. Rivest, *Blind Signatures for Untraceable Payments*, Advances in Cryptology, Proceedings of Crypto 82, pp. 199-203, 1982.

[4] J. Camenisch and M. Stadler. *Efficient group signature schemes for large groups*. In Advances in Cryptology, CRYPTO '97 Proceedings, LLNCS 1294, pages 410–424, Santa Barbara, CA, August 1997.

[5] J. Camenisch and M. Michels. *A group signature scheme based on an RSA-variant*. Tech. Rep. RS-98-27, BRICS, University of Aarhus, Nov. 1998. Preliminary version in ASIACRYPT'98, volume 1514 of LNCS, pages 160–174. Springer Verlag, 1998.

[6] M. Jakobsson, J. P. Hubaux, and L. Buttyan. *A Micro-Payment Scheme Encouraging Collaboration in Multi-hop Cellular Networks*, Financial Cryptography, January 2003.

[7] A. Lysyanskaya and Z. Ramzan. *Group blind digital signatures: A scalable solution to electronic cash*. In Proc. Second International Conference on Financial Cryptography, 1998.

[8] P. Michiardi and R. Molva, *Core: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks*, IFIP - Communication and Multimedia Security Conference 2002.

[9] K.Q. Nguyen, Yi Mu, and V.Varadharajan, *Divertible Zero-Knowledge Proof of Polynomial Relations and Blind Group Signature*, Information Security and Privacy, Proceedings of ACISP'99, April 1999.

[10] Z.A. Ramzan, *Group Blind Digital Signatures: Theory and Applications*, master of science, MIT, 1999.

[11] J.M. Seigneur, S. Farrell, C.D. Jensen, E. Gray, and Y. Chen *End-to-end Trust Starts with Recognition*, in Proceedings of Conference on Security in Pervasive Computing (SPC'2003), Boppard, Germany, March, 2003.