An Anonymous Credential System and a Privacy-Aware PKI

Pino Persiano* and Ivan Visconti *

Dipartimento di Informatica ed Applicazioni
Università di Salerno
Via S. Allende, 84081 Baronissi (SA), Italy
{giuper, visconti}@dia.unisa.it

Abstract. In this paper we present a non-transferable anonymous credential system that is based on the concept of a chameleon certificate. A chameleon certificate is a special certificate that enjoys two interesting properties. Firstly, the owner can choose which attributes of the certificate to disclose. Moreover, a chameleon certificate is multi-show in the sense that several uses of the same chameleon certificate by the same user cannot be linked together.
We adopt the framework of Brands [2] and our construction improves the results of Camenisch *et al.* [5] and Verheul [16] since it allows the owner of a certificate to prove general statements on the attributes encoded in the certificate and our certificates enjoy the multi-show property.

1 Introduction

The protection of private information in access-control based applications is a challenge that can be postponed no longer.

In this work we present new cryptographic techniques to allow the deployment of an anonymous credential system and a privacy-aware PKI that protect both the privacy of the users and the security of the services. The protection of user privacy is based on the disclosure of only the private information strictly necessary for a given transaction. For example, a user that has a credential specifying the year of birth can prove that he was not born in 1974 without disclosing his year of birth. This is achieved by using techniques similar to those of [2] and the result cannot be derived from those of [5, 16] in which one of the following cases happens: 1) the year of the birth must be disclosed; 2) the possession of an ad-hoc credential that states that the year of the birth is different from 1974 is required. In the first case, more information than needed is released since the exact year of the birth is not required. In the second case the user must possess a credential that exactly fits the requirement of the access control policy of the service provider. This last case is not reasonable since an access control policy of a service provider can be short-lived while the process of obtaining a credential

* Supported by a grant from the Università di Salerno and by Young Researchers grants from the CNR.

from a trusted organization requires longer (*e.g.*, when some form of personal identification is required). Moreover, if a user has a credential for each possible access-control based service then the number of credentials he has to deal with becomes impractical.

2 Contributions and Related Work

Related Work. Our work follows the lead of Brands [2] that constructed a certificate system in which a user has control over what is known about the attributes of his certificate. Moreover, within the settings of Brands, it is possible for a user to give interactive or non-interactive proofs that the attributes encoded in a certificate enjoy a given property as encoded by a linear Boolean formula. The main drawback of Brands' certificates is that they are *one-show* in the sense that using the same certificate twice makes the two transactions linkable even though the attributes are still hidden. In the rest of the paper we call *multi-show* a certificate that can be used several times and still guarantees unlinkability. We will base our construction on some techniques proposed in [2] in order to achieve proofs of possession of attributes that satisfy a linear Boolean formula and we extend such schemes in order to achieve the *multi-show* property.

Recently, Verheul [16] proposed a new solution for multi-show digital certificates. In his model, that supports a multi-show property similar to the one that we adopt in this paper, the owner of a certificate can construct by himself another certificate with the same attributes of the original one and such that they are unlinkable. The result is based on the assumption that for some groups the Decisional version of the Diffie-Hellman (DDH) problem is easy while its Computational version (CDH) is hard and on an additional *ad-hoc* assumption. However, Verheul's certificates do not allow the user to prove in a zero-knowledge fashion properties of the attributes of his certificate as in the case of Brands' certificates (see our discussion at the end of Section 1).

Another approach to prove possession of attributes has been addressed as *Anonymous Credentials*. In this approach the user is the owner of some credentials released by trusted organizations. In order to achieve anonymity, credentials should change their aspect so that different proofs of possession are unlinkable. The first implementation of anonymous credentials is presented in [10] where an interaction with a third party is always necessary in order to achieve unlinkability. Lysyanskaya *et al.* [14] proposed a general credential system that, however, is impractical being based on general zero-knowledge proofs. Several papers have then improved on these pioneering work, most notably the work of Camenisch and Lysyanskaya [5] that proposed a practical anonymous credential system that is based on the strong RSA assumption and the DDH assumption. In the system of [5] it is possible to unlinkably prove possession of a credential supporting the multi-show property and the entities that release credentials can independently choose their cryptographic keys. However the approach is subject to the lending problem and the proposed solution based on a new primitive called *circular*

encryption affects intensively the performances and it is not possible to prove possession of attributes that satisfy a given linear Boolean formula.

Contributions of the paper. In this paper we present a system based on the concept of a *chameleon certificate*. A chameleon certificate is a digital certificate similar to the one proposed by Brands in [2] and thus conceptually similar to an X509 v.3 [13] digital certificate. Chameleon certificates enjoy the following two important properties: 1) the owner of a certificate has complete control on the amount of information about its attributes that is released; 2) different uses of the same certificate are unlinkable. The second property is not enjoyed by Brands' certificates, while the first property is not enjoyed by the anonymous credential system of [5] and by the credential certificate system of [16].

Being conceptually similar to X509 v.3 certificates, chameleon certificates can be easily adapted to work in a PKIX-like scenario even though the protocols to be implemented are obviously different. Moreover, we show that it is possible to prove properties of credentials encoded in a chameleon certificate provided that they could be expressed as a linear Boolean formula. Such proof systems were first considered (and shown to exist) in the framework of a general work on zero-knowledge proof systems for Boolean formula by [11].

We will use both the terms of credential and attribute in order to refer to a private information of a user, since the term credential is typically used in credential systems while the term attribute is typically used in a PKI context.

3 Background and assumptions

In this section we summarize the main cryptographic techniques and assumptions that we use in our constructions. For details see [15].

RSA malleability. The RSA signature scheme is multiplicative:

$$m_1^d \pmod n \cdot m_2^d \pmod n \equiv (m_1 \cdot m_2)^d \pmod n.$$

A consequence of this property is that having the signatures of two different messages it is possible to compute the signature of their product without using the private key. Moreover having just one signature it is possible by using exponentiations to compute the signatures of other related messages. The malleability property is sometimes seen as a drawback for security properties but it has been heavily used (*e.g.*, for privacy [9]).

Assumptions. We give now some assumptions that will be used in our constructions.

Definition 1. *The* discrete logarithm problem (DL problem) *is the following: let q be a prime and Z_q^* be the finite cyclic group of the first $q-1$ positive integers. Let g be a generator of Z_q^*, and let $y \in Z_q^*$. Find the unique integer x, such that $0 \leq x \leq q-1$ and $y = g^x \bmod q$.*

Definition 2. *The* discrete logarithm representation problem (DLREP problem) *is the following: let q be a prime and Z_q^* be the finite cyclic group of the first $q-1$ positive integers. Let $g_0, g_1, \cdots, g_l \neq 1$ be elements of Z_q^* and let $y \in Z_q^*$. Find a tuple $(x_0, x_1, \ldots, x_l)$ called* $(q, g_0, g_1, \cdots, g_l)$-representation *of $y \in Z_q^*$ such that $y = g_0^{x_0} g_1^{x_1} \cdots g_l^{x_l} \bmod q$.*

The following proposition states that if the DL problem is hard then the DLREP problem is hard.

Proposition 1. *Given an instance of the DL problem it is possible to construct an instance of the DLREP problem such that if there is an efficient algorithm A that solves with non-negligible probability the DLREP instance then there is another efficient algorithm A' that solves with non-negligible probability the DL instance.*

The following proposition states that the discrete logarithm problem and the discrete logarithm representation problem are hard also with respect to Z_n^* with n composite. For details see [15].

Proposition 2. *Let n be a composite integer. If the discrete logarithm problem in Z_n^* can be solved in polynomial time, then n can be factored in expected polynomial time.*

In [8] Camenisch and Stadler base the security of a group signature scheme on the assumption that, on input an integer $n = pq$ where p and q are primes of the same length, an integer e such that $(e, \phi(n)) = 1$ and $a \in Z_n^*$, it is hard to find in probabilistic polynomial time a pair (v, x) such that $v^e = a^x + 1 \bmod n$. Moreover, in [8] it is assumed that a pair (v, x) is hard to find even if several other pairs are known. This property is used in order to prove the unforgeability with respect to coalitions of users. Subsequently, in [1] the assumption described above has been shown to be fallacious and in [4] a new assumption, sufficient to prove correctness of a modified group signature scheme, is introduced: given an integer $n = pq$ where p and q are primes of the same length, an integer e such that $(e, \phi(n)) = 1$ and two integers $a, c \in Z_n^*$, it is hard to find in probabilistic polynomial time a pair (v, x) such that $v^e = a^x + c \bmod n$.

In this paper we shall use a generalization of the (modified) Camenisch-Stadler assumption to prove some security properties of our scheme.

We introduce now two assumptions that will be used in order to prove unlinkability and unforgeability properties of our construction. The first assumption states that it is not possible for an efficient algorithm on input $g_1, g_2 \in_R Z_n^*$, where $n = pq$ and p, q are primes, to establish if $g_1 \in \langle g_2 \rangle$ (we denote by $\langle g \rangle$ the group generated by g) even if the factorization of n is known.

More precisely, we define *success probability* $\mathrm{Succ}_1^{A_1}(k)$ of a probabilistic algorithm A_1 as the following probability:

$$\mathrm{Succ}_1^{A_1}(k) = Pr((n, p, q, g_1, g_2) \leftarrow \mathrm{GenPrimes}(1^k);$$
$$b \in \{0,1\} \leftarrow A_1(n, p, q, g_1, g_2) : b = 0 \text{ if } g_2 \in \langle g_1 \rangle \text{ or } b = 1 \text{ if } g_2 \notin \langle g_1 \rangle)$$

where GenPrimes is an algorithm that, on input 1^k, outputs two randomly chosen primes p, q of length k, their product n and two randomly chosen elements $g_1, g_2 \in Z_n^*$.

Assumption 1 *For all efficient algorithms A_1, for all constants c and for all sufficiently large k*

$$\mathrm{Succ}_1^{A_1}(k) \leq 1/2 + k^{-c}.$$

For our second assumption, we consider a probabilistic polynomial-time algorithm A_2 that receives as input

1. an integer n such that $n = pq$ where p and q are primes of length k;
2. $e \in Z_n^*$ such that $(e, \phi(n)) = 1$;
3. $g, g_0, g_1, \ldots, g_l \in Z_n^*$;
4. s such that $g = s^e \bmod n$.

and has access to an oracle $\mathcal{O}$ that on input $(x_0, \ldots, x_{l-2})$ outputs $(v, x_0, x_1, \ldots, x_l)$ such that x_{l-1} and x_l are uniformly distributed over Z_n^* and $v^e = g_0^{x_0} g_1^{x_1} \cdots g_{l-1}^{x_{l-1}} + g_l^{x_l} \bmod n$.

We denote by $\mathrm{Succ}_2^{A_2}(k)$ the probability that algorithm A_2, on input randomly chosen $(n, e, g, g_0, \cdots, g_l, s)$ with n product of two primes of length k and having access to $\mathcal{O}$ outputs a tuple $(v, x, y, x_0, x_1, \ldots, x_l)$ such that $v^e = g^x g_0^{x_0} g_1^{x_1} \cdots g_{l-1}^{x_{l-1}} + g_l^{x_l} g^y \bmod n$ and $(x_0, x_1, \ldots, x_l)$ is not part of one of the oracle's replies.

Assumption 2 *For all efficient algorithms A_2, for all constants c and for all sufficiently large k*

$$\mathrm{Succ}_2^{A_2}(k) \leq k^{-c}.$$

We observe that it is very easy, given a tuple $(v, x_0, x_1, \ldots, x_l)$, to output a new tuple $(v', x, x, x_0, x_1, \ldots, x_l)$. Indeed we will use exactly this property in order to achieve unlinkability. However, we stress that in order to break our assumption it is necessary to produce a new tuple in which the sequence $x_0, x_1, \ldots, x_l$ is different from that of each original tuple. We notice that our non-standard intractability assumption is similar to the Camenisch-Stadler one. We are neither aware of any corroboration that it should be hard, nor can we break it. The following three obvious attacks do not seem to work:

1. if the adversary first chooses $x, x_0, x_1, \ldots, x_l$, then, to compute the value v, the adversary has to break the RSA assumption;
2. if the adversary randomly chooses a pair (v, z) such that $z = v^e \pmod n$ then he has to compute two representations with respect to the given bases whose sum is z, and this seems to be an intractable problem;
3. if the adversary uses the malleability of RSA multiplying elements for which he knows the representations and the RSA signatures (as we stated in the assumption), he does not obtain a new valid tuple $(v, x, y, x_0, \ldots, x_l)$.

Brands' results. In [2], a mechanism to prove knowledge of a $(n, g_0, g_1, \ldots, g_m)$-representation of an integer y satisfying a given linear Boolean formula is presented. This is achieved by showing that the knowledge of a $(n, g_0, g_1, \ldots, g_m)$-representation can be used to prove the knowledge of another specific representation if and only if the values satisfy a given linear Boolean formula. In particular Brands' interactive proofs of knowledge are honest verifier zero-knowledge while the non-interactive proofs of knowledge are secure in the random oracle model. We will use these results to guarantee the privacy property of our construction.

Proofs over committed values. In [7] the authors present a proof system for proving that the sum of two committed integers is equal to a third committed integer modulo a fourth committed integer is presented. For details see Section 3 of [7]. We will use this result to guarantee the unlinkability property of our construction.

4 Chameleon certificates

Our model consists of three types of players:

1. The *organizations*, that release *master chameleon certificates* to users.
2. The *users*, each with has a set of attributes and a private key. A user receives a master chameleon certificate encoding his attributes from an organization that he will then use to construct unlinkable *slave* chameleon certificates. Slave chameleon certificates are then used to prove possession of credentials.
3. The *service providers*, that use access control policies in order to protect their resources. Each service provider discriminates between legitimate users of the service and users that do not have the rights to access the service. We assume that the access control policy for each resource of each service provider is represented by a formula Φ over the credentials of the users.

Next, we summarize the procedures executed by the parties in the system for which we are going to present an implementation in the next section.

1. **System set-up**: this step is performed only once by each organization in order to establish publicly verifiable parameters that will be used by the next procedures. At the end of this phase, the organization is ready to release chameleon certificates.
2. **User enrollment**: this step is performed by the user and by an organization. The user asks for a *master* chameleon certificate corresponding to a set of credentials. The organization verifies the credentials and then releases the master chameleon certificate.
3. **Refreshing**: this step is performed by a user that holds a master chameleon certificates in order to obtain a *slave* chameleon certificate that contains the same attributes and public key of the master chameleon certificate but such that the slave and the master chameleon certificates cannot be linked together.

4. **Showing possession of credentials**: this step is performed by a user that interacts with a service provider in order to gain access to a service restricted to legitimate users.

We wish to guarantee the following properties:

1. **Unforgeability**: it is computationally infeasible for a coalition of users to generate a new master chameleon certificate without the help of an organization or to generate a slave (or a master) chameleon certificate whose encoded credentials are different from one of the master chameleon certificates received from an organization.
2. **Unlinkability**: a slave chameleon certificate cannot be linked to the master chameleon certificate or to other slave chameleon certificates.
3. **Privacy**: it is infeasible for a service provider to compute the value of any attribute hidden by a master or a slave chameleon certificate or to gain more information with respect to the one disclosed by the user by proving the satisfaction of a linear Boolean formula.
4. **Malleability**: the refreshing procedure can be executed by the client without interacting with any organization.
5. **Usability**: a slave chameleon certificate can be verified as authentic by the service provider.
6. **Lending**: it is inconvenient for a legitimate user to share its credentials with other users.

Let us discuss the properties listed above. When a user receives the master chameleon certificate from an organization he can construct other certificates (*i.e.*, slave chameleon certificates) such that they are unlinkable (property 2) to the first one (*i.e.*, there is privacy with respect to the organization) and unlinkable among themselves (*i.e.*, there is privacy with respect to a coalition of organizations and service providers that share the issued/received chameleon certificates). In particular the construction of usable slave chameleon certificates can be performed without interacting with other parties (properties 4 and 5), thus it can be distributed over time as the user prefers (*i.e.*, this implies the multi-show property). A master chameleon certificate and its corresponding slave chameleon certificates do not expose directly information stored in them, selective disclosure of user information and satisfaction of linear Boolean formulas is possible, while the construction of a user profile by an organization or a coalition of organizations is hard to perform (properties 2 and 3). A coalition of users cannot construct a new valid chameleon certificate whose credentials are different from the ones encoded in at least one of the released master chameleon certificates (property 1). Of course a user can always give all his private information to another one lending the secrets that correspond to his certificate, so we require that the lending of private credential is inconvenient (property 6).

5 A construction for chameleon certificates

System set-up. For the sake of ease of exposition, we now present our system only for the case in which a chameleon certificate carries two credentials. We

stress that modifying the system in order to support more than two credentials is straightforward.

Organization O performs the following steps:

1. randomly picks a pair $(P_O = (n, e), S_O = (n, d))$ of RSA public and private keys where $n = pq$ and p, q are k-bit prime integers;
2. randomly picks 5 elements $g_0, g_1, g_2, g_3, g_4 \in Z_n^*$ and an element $g \in Z_n^*$ such that the order of g is unknown (*e.g.*, it can be taken from a public random string);
3. computes a signature $s = g^d \bmod n$ of g;
4. publicizes public(O)= $(P_O, g, s, g_0, g_1, g_2, g_3, g_4)$.

The bases g_1, g_2 are used to encode the 2 credentials of a certificate, g and s are used in order to achieve unlinkability, g_0 is used to encode user key, g_3 and g_4 are used against adaptive attacks.

User enrollment. In this phase user U asks to the organization O for a chameleon certificate with encoded values x_1, x_2 of the 2 credentials and the public key $P_u = {g_0}^{x_0}$. The values x_1, x_2 of the credentials are sent in clear to O while the secret key x_0 is kept secret and only P_u along with a proof of knowledge of its discrete logarithm with respect to the base g_0 is sent to O.

Once the attributes of the user have been verified in accordance to the policies of the organization, O randomly chooses $x_3, x_4 \in Z_n^*$ and releases a master chameleon certificate that consists of the pair (C, S) where

$$C = P_u g_1^{x_1} g_2^{x_2} g_3^{x_3} + g_4^{x_4} \bmod n \text{ and } S = C^d \bmod n.$$

The user U receives $(C, S), x_3, x_4$ and verifies that the master chameleon certificate has been correctly computed.

Refreshing. Now we present our *refreshing* procedure that is executed by the user each time he needs to exhibit a slave chameleon certificate. Starting from a chameleon certificate (C, S) a new slave chameleon certificate is generated by the user in the following way:

1. pick a random value $x \in Z_n^*$ and computes $C' = g^x \cdot C \bmod n$;
2. compute a signature S' of C' as $S' = s^x \cdot S \bmod n$;
3. the slave chameleon certificate is (C', S').

Showing possession of credentials. In this phase a user proves to a service provider the possession of a master chameleon certificate (C', S') in order to obtain access to a service. The access control policy of the service provider for a given resource is described by a linear Boolean formula Φ and the user proves that the credentials encoded in the master chameleon certificates satisfy the formula Φ. More precisely, the following steps are performed by the user and the service provider.

1. The user generates a slave chameleon certificates (C', S') by picking a random x and setting $C'_0 = g^x g_0^{x_0} g_1^{x_1} g_2^{x_2} g_3^{x_3}$ and $C'_1 = g_4^{x_4} g^x$ (so that $C' = C'_0 + C'_1 \pmod n$);
2. The user computes commitments $(\hat{C}'_0, \hat{C}'_1, \hat{C}')$ of C'_0, C'_1, C' using the techniques of [7] and sends them to the service provider.
3. The service provider sends $b \in_R \{0, 1, 2\}$ as challenge.
4. If b is 0 then the user proves that $(\hat{C}'_0, \hat{C}'_1, \hat{C}')$ are well computed, i.e., $\hat{C}'$ is the commitment of the sum modulo n of two values whose commitments are $\hat{C}'_0$ and $\hat{C}'_1$ and that $\hat{C}'$ is the commitment of C'. This is achieved by using the proof systems described in [7]. Moreover the user sends S' and the service provider verifies that S' is a correct signature of C'.
5. If b is 1 then the user opens $\hat{C}'_0$ and both parties engage in a PoK in which the user proves to know a $(n, g, g_0, g_1, g_2, g_3)$-representation (x, x_0, x_1, x_2, x_3) of C'_0 such that $\Phi(x, x_0, x_1, x_2, x_3) = 1$. This is achieved by using the results described in [2].
6. If $b = 2$ then the user opens $\hat{C}'_1$ and proves that it knows the (n, g, g_4)-representation (x, x_4) of C'_1. This is achieved by using the proof of knowledge of a representation.

Notice that only the owner of the certificate knows the (n, g, g_4)-representation of C'_1 and the $(n, g, g_0, g_1, g_2, g_3)$-representation of C'_0, thus only the legitimate owner of the certificate can use the certificate, since it is the only party that knows the private key x_0.

The steps described above must be repeated several times in order to gain a satisfying soundness. At each iteration a new slave chameleon certificate is used.

Security of the system. We now discuss the security of our proposal with respect to the properties that we described in Section 4.

Unforgeability. A coalition of users can share the secrets of their master/slave chameleon certificates in order to obtains a new master/slave chameleon certificate whose attributes $x_0, x_1, \ldots, x_4$ are different with respect to any shared master/slave chameleon certificate. However if such an algorithm exists, another algorithm that uses the first one and exploits the malleability property of RSA can easily break Assumption 2.

Unlinkability. Suppose that there exists an algorithm A that guesses whether a given slave chameleon certificate (C', S') is related to a given master chameleon certificate (C, S). More precisely, algorithm A receives as input the information publicized at set-up phase by O along with the transcript of a transaction in which the slave chameleon certificate has been used. A has to distinguish between two cases: (C', S') has been obtained by running the refreshing procedure on input master chameleon certificate (C, S) or by running the refreshing procedure on input a randomly chosen master chameleon certificate (C^*, S^*). We say that the adversary succeeds if it has probability of guessing correctly significantly better than 1/2. We show that A can be used to break Assumption 1. Consider now an algorithm A' that receives two primes p, q of the same length

and $y, g \in Z_n^*$ where $n = pq$ and has to output a guess to whether $y \in \langle g \rangle$. A' generates an RSA key $((n,d),(n,e))$, randomly chooses $g_0, \ldots, g_l, x_0, x_1, x_2, x_3, x_4$, computes a master chameleon certificate

$$(C'' = g_0^{x_0} g_1^{x_1} g_2^{x_2} g_3^{x_3} + g_4^{x_4} \bmod n, S'' = C''^d \bmod n)$$

and runs

$$A((C'', S''), (C'' \cdot y, S'' \cdot y^d \bmod n), (n, d, e, p, q, g_0, g_1, g_2, g_3, g_4, x_0, x_1, x_2, x_3, x_4)).$$

If the output of A is `true` then A' can establish that $y \in \langle g \rangle$ else $y \notin \langle g \rangle$. Thus A contradicts Assumption 1.

A similar result can be given for the case of the linking of two slave chameleon certificates.

Privacy. Using the interactive honest-verifier ZKPoK or the non-interactive PoK in the random oracle model of the $(n, g, g_0, g_1, g_2, g_3)$-representation of the first component of a slave chameleon certificate and the proof of knowledge of satisfaction of linear Boolean formula over its attributes presented in [2] we have that the selective disclosure property of our scheme holds.

Malleability. Each step of the refreshing procedure can be executed by the user without interacting with any party. Moreover such operations can be performed in any moment since data extracted from an on-line interaction are not required.

Usability. It is easy to verify that by the malleability property of RSA signatures, in the output of the refreshing procedure (C', S') on input a master chameleon certificate (C, S), S' is a valid RSA signature of C'. We stress that this procedure does not require interaction with any party.

Lending. When a user shares his private information regarding a chameleon certificate with other users then they can use the certificate for their purposes since there is a complete sharing of user identity. Even if the presence of the private key is a first deterrent to this drawback a more sophisticate strategy to discourage such sharing can be achieved by adding attributes that typically are not shared by their owners. For example another base $g_c \in Z_n^*$ could be considered and inserted in each certificate to represent a credit card number. Using this mechanism each user that tries to use such a master chameleon certificate or one of its corresponding slaves needs to know the owner's credit card number to convince the verifier during the PoK.

5.1 Applications

Based on the concept of a chameleon certificate, we can design a system for non-transferable anonymous credential system including the following parties:

1. the organizations that release credentials;
2. the users that get the credentials and give proofs of possession;
3. the service providers that trust the organizations and restrict their services to the users that possess some credentials.

Each organization that releases credentials publicizes the list of *supported* credentials. The credentials are released by encoding them in chameleon certificates and thus the corresponding public information are publicized too.

Each service provider publicizes the list of trusted organizations. Moreover for each restricted resource there is a corresponding linear Boolean formula over some credentials. The service provider knows the list of credentials released by each of his trusted organizations and thus the case that a linear Boolean formula refers to credentials that are not totally released by at least a trusted organization cannot happen.

A user needs at least one master chameleon certificate released by one organization. However the organizations do not necessarily grant on the same credentials, in this case a user could receive some master chameleon certificates from different organizations. Moreover the service providers do not necessarily trust the same organizations and thus the same credentials could be repeated in different master chameleon certificates so that the right one is selected during user enrollment.

In order to prove possession of some credentials, the user performs the *refreshing* procedure on the master chameleon certificate in order to obtain the slave chameleon certificates (this step can be executed off-line). Then the user proves the possession of credentials that satisfy the linear Boolean formula that corresponds to the requested resource as we discussed in Section 5.

Privacy-aware PKI. A privacy-aware PKI is obtained by using chameleon certificates. The role of organizations is played by the certification authorities. The credentials encoded in the certificate are the attributes that are assigned to the owner. The master chameleon certificate (C, S) can be publicized along with the two shares C_0 and C_1 such that $C_0 + C_1 = C \bmod n$ and such that the owner knows the representations of C_0 and C_1 with respect to the appropriate bases.

Certificate revocation. The revocation of a chameleon certificate is possible in different ways. Following the approach of Verheul it is possible to use short-lived certificates. Another possibility is to use standard CRLs represented by sequences of serial numbers, in this case the serial number can be encoded as an attribute of the chameleon certificate and during their use it is necessary to prove that the corresponding attribute is different from each serial number contained in a CRL. Of course when the size of the CRL increases the performances of the protocols decrease. Finally, we can use the general technique of [6] to make revocable a chameleon certificates.

The PKI discussed above is similar to the one proposed in [2], but in our case the owner of a certificate can use it for anonymous identification or to prove possession attributes in such a way that different transactions are not linkable.

6 Conclusions

In this paper we have introduced the concept of a *chameleon certificate*. Moreover, we have presented a construction for chameleon certificates that is based

on a generalization of the assumption of [8]. We have shown a non-transferable anonymous credential system and a PKI based on chameleon certificates. Finally, we remark that the proof systems of [3] can also be used in our construction.

References

1. G. Ateniese and G. Tsudik, Some open issues and new directions in group signatures. *Financial Cryptography 1999*, volume 1648 of LNCS.
2. S. Brands, *Rethinking Public Key Infrastructures and Digital Certificates; Building in Privacy*. MIT Press, 2000.
3. E. Bresson and J. Stern., Proofs of knowledge for non-monotone discrete-log formulae and applications. In *Proceedings of International Security Conference (ISC 2002)*, volume 2433 of LNCS.
4. J. Camenisch., A note on one of the assumptions. Jan Camenisch Home Page - Selected Publications.
5. J. Camenisch and A. Lysyanskaya. An efficient non-transferable anonymous multishow credential system with optional anonymity revocation, *Eurocrypt 2001*, volume 2045 of LNCS.
6. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. *Crypto 2002*, volume 2442 of LNCS.
7. J. Camenisch and M. Michels. Proving in zero-knowledge that a number is the product of two safe primes. *Eurocrypt 99*, volume 1592 of LNCS.
8. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. *Crypto 97*, volume 1294 of LNCS.
9. D. Chaum. Blind signatures for untraceable payments. *Crypto 82*.
10. D. Chaum and J. Evertse. A secure and privacy-protecting protocol for transmitting personal information between organizations. *Crypto '86*, volume 263 of LNCS.
11. A. De Santis, G. Di Crescenzo, G. Persiano, and M. Yung. On monotone formula closure of SZK. FOCS 1994.
12. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. *Crypto '86*, volume 263 of LNCS.
13. R. Housley, W. Polk, W. Ford, and D. Solo. Internet X509 public key infrastructure: Certificate and Certificate Revocation List (CRL) Profile. Network Working Group, RFC 3280, April 2002.
14. A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. *Proceedings of Selected Areas in Cryptography* 1999, volume 1758 of LNCS.
15. A. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.
16. E. Verheul. Self-blindable credential certificates from the weil pairing. *ASIACRYPT 2001*, volume 2248 of LNCS.