Towards a Privacy-preserving National Identity Card

Yves Deswarte, Sébastien Gambs

Towards a Privacy-preserving National Identity Card

Yves Deswarte[1,2], Sébastien Gambs[1,2]

[1] CNRS ; LAAS ; 7 avenue du Colonel Roche, F-31077 Toulouse, France
[2] Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France
{Yves.Deswarte,Sebastien.Gambs}@laas.fr

Abstract. In this paper, we propose to replace the national identity card, currently used in many countries, by a personal device that allows its user to prove some binary statements about himself while minimizing personal information leakage. The privacy of the user is protected through the use of anonymous credentials which allows him to prove binary statements about himself to another entity without having to disclose his identity or any unnecessary information. The proposed scheme also prevents the possibility of tracing the user, even if he proves several times the same statement (unlinkability property). A tamper-proof smartcard is used to store the personal information of the user thus protecting his privacy and preventing the risks of forgery at the same time. The user identifies himself to the card via biometrics thus forbidding an unauthorized use in the situation where the card is stolen or lost. Two practical implementations of the privacy-preserving identity card are described and discussed.

1 Introduction

Intuitively respecting the principles of *data minimization*[1] and *data sovereignty*[2] when using a national identity card seems to be at odds with other obligations required in practical tasks from everyday life such as checking the nationality of the owner of the card when he crosses a border, verifying his age when he wants to obtain some discount related to it or proving that he belongs (or does not belong) to a particular group. In this paper, we advocate that this intuition is wrong by introducing the concept of *privacy-preserving identity card*.

[1] The data minimization principle states that only the information necessary to complete a particular application should be disclose (and no more). This principle is a direct application of the legitimacy criteria defined by the European data protection directive (Article 7, [14]).

[2] The data sovereignty principle states that the data related to an individual belong to him and that he should stay in control of how these data are used and for which purpose. This principle can be seen as an extension of many national legislations on medical data that consider that a patient record belongs to the patient, and not to the doctor that creates or updates it, nor to the hospital that stores it.

Definition 1 (Privacy-preserving Identity Card) *A* privacy-preserving identity card *is a personal device that allows its user*[3] *to prove some binary statements about himself (such as his right of access to some resources) while minimizing personal information leakage.*

Consider for instance the following scenario that illustrates how such a card would work in practice.

Scenario 1 (Alice in Anonymityland) *Alice is privacy addicted since she has read the seminal paper of Chaum on anonymity [10]. She has recently seen in an advertisement that her government now offers the possibility of using a privacy-preserving identity card. Therefore, she goes to the town hall and asks for it. The city hall checks the validity of Alice's identity, scans her biometric data and sends them in a secure manner (for instance using a protected conveyor) along with her personal information to the corresponding governmental service that will be responsible for issuing the card.*

For an external observer, the card looks exactly the same as any other privacy-preserving identity card, since there is no personal information written on the plastic card. Effectively, the card is a tamper-proof smartcard containing anonymous credentials that Alice can use to prove some statements about her. The card is activated by Alice's biometric features. For instance, her card allows Alice to prove her nationality when she crosses the border, to show that she is within some age interval in order to gain some discount at the theater, to certify her identity when she takes the plane or to gain access to local services restricted to her neighborhood residents.

If the card of Alice is lost or stolen, she does not need to worry about this misuse for malicious purpose, because of the biometrics authentication and the tamper-proof features of the smartcard. Instead, she simply returns to the city hall to declare the loss of her card and ask for a fresh privacy-preserving identity card.

Our proposal for the privacy-preserving national identity card is close in spirit to the project PRIME[4] (PRivacy and Identity Management for Europe) [20], whose goal was to develop a framework and tools allowing a user to manage his identity and to protect his privacy in the cyberspace. Indeed, the main purpose of the privacy-preserving identity card is to enable a person to conduct tasks in the real world without having to disclose his identity whereas PRIME was focusing exclusively on the online setting. The informal proposition of Birch for the future U.K. national identity card [4], called Psychic ID, also shares several privacy features with our proposal. Indeed, the Psychic ID card respects the principle of data minimization and only reveals to a reader (or visually to an entitled person) the minimal information concerning the user that is needed for

[3] In this paper, we use the word "user" to denote at the same time both the owner and the effective user of the card. Indeed as the user needs to authenticate to the card before he can use it, the user of the card will effectively always be also his owner.

[4] `https://www.prime-project.eu/`

a specific purpose if the user possesses the corresponding credential, and nothing otherwise. An overview of the privacy features of the specifications of the future European electronic identity cards can be found in [15]. Other related works close to our approach include a protocol for the partial revelation of information related to certified identity proposed by Boudot [5] and the development of a cryptographic framework for the controlled release of certified data due to Bangerter, Camenisch and Lysyanskaya [1].

The outline of the paper is the following. First in Section 2, we detail in an abstract way the desirable properties that a privacy-preserving identity card should fulfill. Afterwards in Section 3, we briefly review some enabling technologies on smartcards, anonymous credentials and biometric authentication that will be the basis of our practical implementations of the card. Then in Section 4, we briefly describe of such a card might be use in practice before in Section 5 and Section 6, proposing two practical implementations of the privacy-preserving identity card. Finally, we conclude in Section 7 with a discussion on possible extensions to the privacy-preserving identity card.

2 Desiderata for a Privacy-preserving Identity Card

In this paper, we adopt a notation inspired from the work of Camenisch and Lysyanskaya on *anonymous credentials* [7] (see Section 3.2 for more details). In particular, we call the owner of the privacy-preserving identity card, the *user* (who is likely to be a simple citizen such as Alice). The *Registration Authority* (RA) is a legal entity (such as the city hall) that can check the personal information of the user and register the request for a privacy-preserving identity card. The *Certification Authority* (CA) is a trusted third party (for instance the government) that will sign the information transmitted by the RA to certify its validity. Once the card has been issued, the RA and CA are no longer involved in the picture except if the user needs a new card or if there is a valid reason for lifting the anonymity of the user. An *organization* is an entity that can grant access to some of its resources to the user. (For example, in Scenario 1 an organization could be the immigration services, the theater or an airline company.) A *verifier* belongs to one organization and interacts with the user to check his right of access to the resources of this organization. In practice, the verifier is usually a smartcard reader device connected to the network of the organization that can communicate with the privacy-preserving identity card.

As illustrated in Scenario 1, ideally the privacy-preserving identity card should fulfill the following properties:

- *No personal information leakage*: in order to protect the privacy of the user, the card should disclose as little information as possible about him. Ideally, the only thing the card should reveal is one bit of information proving (or disproving) a binary statement concerning the user.
- *Unlinkability*: it should not possible to trace and link the actions of the user of the card. For instance, even if the user proves the same statement at

different occasions, it should be impossible to link the different statements as being made by the same user.
- *Ownership proof*: only the legitimate user should be able to use his privacy-preserving identity card to prove statements about himself to other entities. This means that some authentication mechanism has to take place between the user and the card. The purpose of this authentication step is to avoid an unauthorized use of the card. This authentication mechanism should also guarantee the *non-transferability* of the card. Otherwise, the user could sell for some money the use of his privacy-preserving identity card to somebody else, thus transferring his privileges or even his identity to illegitimate users.
- *Authenticity*: some mutual authentication has to be performed between the card and the reader device in order to prevent the possibility of an adversary impersonating the role of a valid privacy-preserving identity card or a valid reader. This authentication will involve two-way communication between the identity card and the reader and will assert the authenticity of both the card and the reader.
- *Correctness*: a binary statement proven by the user with the help of the privacy-preserving identity card should always[5] be valid. For instance, the user should never be able to prove false statements about himself by cheating the system (*soundness property*). Moreover if the verifier is honest, he should always accept a binary statement about the user provided that this statement is true and the user possesses the corresponding credentials (*completeness property*).
- *Unforgeability*: in order to avoid someone counterfeiting the identity card and usurping the role of the user, the card should be tamper-proof and have an inherent ability to resist hardware and logical attacks.

Apart from these fundamental requirements, the privacy-preserving identity card may also respect some additional properties such as:

- *Optional anonymity removing*: the actions of the user should stay anonymous at all times, except in some scenarios where it might be necessary to remove his anonymity for serious reasons. For instance in an extreme situation, it could happen that a crime (such as a murder) has been perpetrated in a room that has been accessed by only one person using a privacy-preserving identity card. In this situation, the certification authority and the verifier may want to collaborate in order to lift the anonymity of this person. On the other hand, although the possibility of lifting the anonymity is desirable in some scenarios, it could decrease the confidence of the user in his belief that his privacy will really be protected by the card.
- *Explicit consent*: in order to increase the trust of the user in the system, the card could monitor the questions that it has been asked and display them

[5] In this paper when we use the terms "always" or "never", it means respectively that this event occurs or does not occur, except with negligible probability (for instance exponentially small with respect to a security parameter). In the same manner, in all this paper we consider that an adversary has only bounded computational power.

to the user. It is even possible to imagine, that for some questions that are deemed critical regarding the privacy of the user, his confirmation may be asked before the privacy-preserving identity card replies to the question.

3 Enabling technologies

Enforcing in reality the properties of the privacy-preserving identity card require the combination of several hardware and cryptographic techniques that we briefly review in this section.

3.1 Smartcards

A *smartcard* is a plastic card with an embedded integrated circuit that contains some dedicated memory cells and a microprocessor that can process data stored in the memory cells or exchanged with a reader through serial link connections (for contact smartcards), or through radio links (for contactless smartcards). The memory cells can only be accessed by the microprocessor. The main purpose of the smartcard is to assure the confidentiality and integrity of the information stored on the card. For that, the smartcard must satisfy inherent tamper-proof properties (to protect the microprocessor and the memory) as well as some resistance against physical attacks and side-channel analysis[6] [2]. As in cryptology, there is a ongoing race in the smartcard world between the developers of attacks and the designers of counter-measures (see [21] for instance).

Nowadays, smartcards are widely used around the world, especially in mobile phones, tokens for public transport systems or for other applications such as electronic payments. Until now, smartcards used in practice have relied mostly on symmetric encryption (by using algorithms such as triple DES or CRYPTO-1)[7]. Calmels, Canard, Girault and Sibert have recently suggested to move instead to asymmetric encryption in the future for RFID tags, both for security and practical reasons [6]. They have also described a low-cost version of a group signature scheme thus demonstrating that such cryptographic primitive are within the reach of inexpensive smartcard technologies.

3.2 Anonymous Credentials

An *anonymous credential* is a cryptographic token which allows a user to prove statements about himself anonymously to verifiers anonymously. Anonymous credentials are generally based on *zero-knowledge* type of proofs [16] and enable the user to prove his accreditation to the verifier without revealing any additional information (such as his identity). The first system of anonymous

[6] The same kind of tamper-proofness techniques can be applied to USB keys, smartcard readers or other hardware devices for similar purposes.

[7] This assertion is true at least for low-cost smartcards, even if public-key cryptosystems are available on many more smartcards, including JavaCards.

credential is due to Chaum [10] and is based on the idea that each organization might know the same user by a different *pseudonym*. The organizations cannot combine their data on a particular user because they are unable to link two different pseudonyms to the same person. *Private credentials* can be derived from other credentials and used to prove relationships between credentials/attributes/organizations without having the risk of linking the different pseudonyms. For instance, Alice might want to prove anonymously to the public transport company that she is a student in order to get a discount on monthly travel fees. In this example, Alice wants to transfer her credentials granted by the university (organization A) to the public transport company (organization B) without revealing her identity.

Credentials can be *one-show* (as it is the case for e-cash) or *multiple shows*. When a user shows multiple times the same credential, this raises the concern of linkability if several actions can be traced to a unique user (even anonymous). One possibility for preventing this is to issue multiple one-show credentials to the same user. Another solution is to use a *group signature scheme* which allows multiple-show unlinkability. Group signature schemes [11] have been introduced by Chaum and van Heyst to provide anonymity to the signer of the message. For that, there is a single public verification key for the group, but each member of the group receives a different private signing key from the group manager (who could be for instance the CA). A group signature scheme (with optional anonymity removing) consists in general of the four following operations:

- *Registration of the user*. During the Join operation, the CA assigns to the user a new private signature key, which we denote by SKG_U.
- *Signature of a message in behalf of the group*. The SignGroup operation takes as input a message m and signing key SKG_U and produces a signature $\sigma_{G,U}(m)$ on this message.
- *Verification of a group signature*. The VerifySignGroup operation allows to check the validity of a group signature. It requires as input a verification key for the group VKG, which has been setup by the CA and is publicly known, as well as a message m and a group signature on this message $\sigma_{G,U}(m)$. VerifySignGroup produces as output either accept or reject depending on the validity of the signature.
- *Anonymity removing*. From the point of view of the verifier, it is impossible to distinguish if two group signatures come from the same individual or not. However in exceptional situations, the CA can (in association with the verifier) retrieve the identity of a particular signer via the LiftAnonymity operation. This operation takes as input a message m and a group signature on this message $\sigma_{G,U}(m)$ and produce as output the identity of the signer U. In practice, this is often done by first finding the corresponding signature key SKG_U and then retrieving the identity associated to this key.

Another possibility for implementing anonymous credentials is to use a *non-interactive zero-knowledge proof* [3] in combination with a *commitment scheme*. A commitment scheme is characterized by two operations:

– *Commitment phase.* During this phase, the Commit operation takes as input a value a and some auxiliary information aux (which corresponds generally to some form of randomness) and produces $comm(a)$ which is a commitment to this particular value a.
– *Opening phase.* The Open operation takes as input a commitment $comm(a)$ and some auxiliary information aux and reveals as output a, the committed value.

A commitment scheme is *perfectly binding* if there is only one a that corresponds to a particular commitment $comm(a)$ (i.e., an adversary cannot open a commitment to several values), and *computationally hiding* if an adversary with bounded computational power cannot open a particular commitment without access to the auxiliary information. Suppose that a prover stores a particular value a and the CA's signature on it, $\sigma_{CA}(a)$, which certifies its validity. The prover may want to show that this value respects a particular binary statement f to a verifier in a zero-knowledge manner. To realize that, the prover sends to the verifier $comm(a) \leftarrow \mathsf{Commit}(a, aux)$, which is a commitment to the value a. Then, the prover issues $\pi \leftarrow \mathsf{Prove}((a, \sigma_{CA}(a), aux)|\mathsf{VerifySign}(a, \sigma_{CA}(a), VK_{CA}) = \mathsf{accept} \wedge a = \mathsf{Open}(comm(a), aux) \wedge f(a) = true)$, which is a non-interactive zero-knowledge proof that the prover knows (a, $\sigma_{CA}(a)$,aux) such that (1) $\sigma_{CA}(a)$ is a valid signature of the CA on a (verified by VK_{CA}, the public verification key of the CA); and (2) the committed value of $comm(a)$ is effectively a; and (3) the value a respects the binary statement f.

3.3 Biometric Authentication

The *biometric profile* of a person is composed of a combination of some physical features that uniquely characterize him. For instance, a biometric feature can be a fingerprint or a picture of the iris. The biometric data of an individual is a part of his identity just as his name or his address. As such biometrics can be used for the purpose of *identification* (i.e., identifying a particular individual in a list of registered people) or *authentication* (verifying that the person claiming an identity is indeed the one who has been registered with this identity).

In order to verify that an individual corresponds to some registered biometric profile, a fresh sample of his biometric data is generally taken and compared with the stored template using a matching algorithm. The matching algorithm computes a dissimilarity (or distance) measure[8] that indicates how far are the two biometric samples. Two biometric samples are considered to belong to the same individual if their dissimilarity is below some well-chosen threshold, which is dependant of the natural variability within the population. A good biometric strategy tries to find a compromise between false acceptance rate or FAR (wrongly recognizing the individual as a particular registered user) and the false rejection rate or FRR (being unable to recognize the registered user). An example of biometric data is the picture of the iris that can be transformed/coded

[8] The dissimilarity measure used can be for instance the Hamming distance, the set difference or the edit distance.

into a vector of 512 bytes called the IrisCode. Afterwards, it is fairly simple to evaluate the dissimilarity between two codewords simply by computing the Hamming distance between these two vectors. In practice, this method can lead to very low rates of false acceptance ($< 0.1\%$) and false rejection ($< 1\%$).

As the biometric features of an individual is an inherent part of his identity, several techniques have been developed to avoid storing explicitly the biometric profile while keeping the possibility of using it for authentication. For instance, the main idea of *cancellable biometrics* [22] is to apply a distortion to the biometric template such that (a) it is not easy to reconstruct the original template from the distorted version and (b) the transformation preserves the distance between two templates. Other techniques have been proposed which combine the use of error-correcting codes and hash function such as the *fuzzy commitment scheme* [18]. Let b be the biometric profile of the user. For the sake of clarity, we assume that b can be represented as a binary vector of length n (i.e., $b \in \{0,1\}^n$)[9]. An error-correcting code C of size n is chosen such that it can correct up to t errors where t is chosen empirically so as to lead to a good trade-off between the FAR and the FRR. A hash function h is also used by the protocol. The fuzzy commitment scheme [18] can be applied to biometric authentication on a smartcard, with two operations:

- *Enrollment phase*. During this phase, the biometric template b is measured through the Enroll operation. A codeword c is drawn uniformly at random from C and $z = c \oplus b$ is computed. The hashed version of this codeword $h(c)$ as well as z are stored on the card.
- *Verification phase*. When the card wants to verify that the user is the owner of the card, the biometric sensor device[10] measures a fresh biometric sample b' and sends it to the card at the beginning of the Verify operation. Afterwards, the card computes $z \oplus b'$ and decodes this towards the nearest codeword c'. The card then calculates $h(c')$ and accepts the user if $h(c') = h(c)$ and rejects him otherwise.

In the same spirit as the fuzzy commitment scheme, a cryptographic primitive known as *fuzzy extractor* has been developed in the recent years (see for instance the survey [13]). This primitive allows to extract a uniformly distributed random string $rand \in \{0,1\}^l$ [11] from a biometric template b in a noise-tolerant manner such that if the input changes to some b' close to b (i.e. $dist(b,b') < t$), the string $rand$ can still be recovered exactly. When initialized for the first time, a fuzzy extractor outputs a helper string called $p \in \{0,1\}^*$, which will be part

[9] In practice, this might not be true when the matching of templates relies on geometric information (for instance in fingerprints), in which case the error-correcting approach has to be adapted to this situation.

[10] Such a sensor can be implemented on the privacy-preserving identity card itself, but in practice it may also be part of the smartcard reader device.

[11] In the basic version l, the length of the random string generated, is smaller than n, the length of the biometric profile. However, this is not really a problem as it is possible to use $rand$ as a seed of a good pseudorandom number generator to generate an almost uniformly random string of arbitrary size.

of the input of subsequent calls to the fuzzy extractor in order to help in reconstructing *rand*. The string p has the property that it can be made public without decreasing the security of *rand*. Formally, a fuzzy extractor consists of two operations:

- *Generation phase*. During the first use of the fuzzy extractor, the operation Generate takes as input a biometric template b and produces as output a uniform random string *rand* and a helper string p.
- *Retrieval phase*. The operation Retrieve takes as input a biometric profile b' which is close to the original profile b (i.e. $dist(b, b') < t$) as well as the helper string p and produces as output the random string *rand*.

One application of fuzzy extractors is the possibility of using the biometric input of the user as a key to encrypt and authenticate the user's data. For instance, *rand* can act as an encryption key which can be retrieved only by the combination of the user's biometric profile and the helper string. As *rand* is never explicitly stored and the user's biometrics acts as a key, this guarantees that only if the correct biometric template is presented, the record of the user can be decrypted.

4 Operation and Use of the Privacy-preserving Identity Card

We suppose that the privacy-preserving identity card is a contact smartcard that has sufficient resistance against physical and logical attacks [12] (see Section 3.1). The smartcard contains a processor that can compute efficiently cryptographic primitives such as asymmetric encryption and group signature verification. The card memory stores identity data similar to those printed on existing identity cards (e.g., names, date and location of birth, address, etc.), plus biometric data and other security-related information, such as public and private keys.

When the smartcard is inserted into a reader device, the smartcard processor initiates a *mutual authentication* between the card and the reader. This phase involves using a group signature and a simple signature scheme. First, the card generates dynamically a pair of public and secret keys for the session and sends the public key along with the corresponding signature (obtained by using its private signature key of the group) to the reader. The reader checks the validity of the signature on the session public key by using the verification key of the group signature. This session public key will be used by the reader to encrypt the information it sends to the card during this session. Afterwards, the reader proves to the card that it is an authentic reader by showing that it possesses a valid credential signed by the CA. The questions that the reader is allowed to ask to the card are also part of this credential, as well as the public key of the reader that the card will use afterwards to encrypt the information it sends to the reader. The reader also sends a random challenge to the card. Then, the

[12] We also assume that the smartcard reader device that will interact with the privacy-preserving identity card possesses similar tamper-proof properties.

card proves to the reader that it is an authentic card by showing that it belongs to the group of valid smartcards/users, more precisely, by signing the random challenge issued by the reader with its private signature key (see Section 3.2).

If the mutual authentication fails, the smartcard is not activated (i.e., its processor does nothing). Contrarily, when the mutual authentication succeeds, the embedded processor initiates a *biometric verification* of the user, by using for instance the fuzzy commitment scheme for biometric authentication described in Section 3.3. Finally, when the biometric authentication is successful, the processor initiates a *question-response* protocol with the reader device. In pratice, the question of the reader could be any binary query related to an attribute of the user such as:

- "Is the user a Finnish citizen?" (for instance when crossing the border),
- "Is the user under 18 years old?" (when proving that the user is within some age interval),
- "Is the user firstname Alice?" (when checking the identity before boarding a plane) or
- "Is the user an inhabitant of Toulouse?" (when accessing a local service restricted to municipality residents).

The question could also concern a combination of different attributes of the user (for instance "Is the user under 18 years old AND an inhabitant of Toulouse?"). If the question-response protocol is implemented through an anonymous credential system that is expressive enough to prove any combination of the logical operations AND, OR and NOT regarding the attributes of the user then it is possible in principle to check any particular binary statement regarding his identity[13]. Note that in any case, the card discloses no personal data, only a binary statement on data provided by the reader, i.e., data that already exist out of the card. For instance, for checking the first name Alice, this information must be sent by the reader to the card, either because the user has claimed it, or because it has been read on another document such as a boarding pass. The card signs the answer to the question and returns it encrypted with the public key of the reader, in such a way that it forbids a potential eavesdropper to learn this answer or to tamper with it.

5 Basic Implementation

The first implementation of the privacy-preserving identity card that we proposed combines the different technologies and concepts briefly reviewed in Section 3. We call it BasicPIC, which stands for Basic implementation of a Privacy-preserving Identity Card (PIC). In this implementation, we suppose that the smartcard tamperproofness is "sufficient". In practice however, it is quite likely that if an adversary spends enough resources and time he will be able to break

[13] See for instance [9] for an efficient implementation of anonymous credentials that allowed to prove AND, OR and NOT statements regarding the attributes encoded.

the tamper proof characteristics of the smartcard and read and/or modify the information stored on it. We address this issue by proposing a more complex implementation, called ExtendedPIC, in the next Section.

5.1 Initialisation

When the user wish to acquire a new privacy-preserving identity card, he goes to an Registration Authority (RA) who can verify the personal data of the user and register the demand. We denote by $a_1, \ldots, a_k$, the k attributes of the user that embodies his identity. For instance, the i^{th} attribute a_i could be a name (string of characters value), a year of birth (integer value) or an address (mix of strings of characters and integers). After having checked the identity of the user, the RA scans the biometric profile of the user b (which could be for instance his fingerprints, the map of his iris or the template of his voice) and computes $h(c), z \leftarrow \mathsf{Enroll}(b)$, where $z = b \oplus c$ for c a random codeword of C and $h(c)$ a hashed version of it. The RA sends z and $h(c)$ in a secure manner along with the personal information of the user to the Certification Authority (CA). The secure transmission of the personal information of the user between the RA and the CA is done by communicating over an electronic secure channel or via a physical delivery whose process is under strict monitoring.

The CA is responsible for issuing the privacy-preserving identity card. The CA performs the Join operation (see Section 3.2) to generate the signing key SKG_U of the user for the group signature. This key is stored within the tamper-proof smartcard that is the core of the privacy-preserving identity card. The attributes of the user $a_1, \ldots, a_k$ as well as z, $h(c)$ and VK_{CA} (the public verification key of the CA) are also stored inside the card. For an external observer, the card is "blank" and looks exactly the same as any other privacy-preserving identity card. The exact form of the smartcard can vary, depending on the chosen trade-off between the individual cost of each card that we are willing to spend and the assumptions we make on the time and means that the adversary is able to deploy. If the technology is affordable, the card could possess a biometric sensor[14] and a screen. The screen could display for instance the identifier of the reader and the questions asked to the card.

Before an organization can use a reader device able to interact with privacy-preserving identity cards, the organization needs first to register the device to the CA. The CA then emits a credential cr in the form of "This reader is allowed to ask the question f to a privacy-preserving identity card. The answer to this question has to be encrypted using the public encryption key EK_R.". The public encryption key EK_R is supposed to be specific to the reader and as such can be considered as its identifier. The CA will certify this credential by performing $\mathsf{Sign}(cr, SK_{CA})$ which generates $\sigma_{CA}(cr)$, the signature on the credential cr using the CA secret key. The reader also knows the group verification key VKG

[14] Some companies, such as Novacard, have started to sell smartcard integrating a fingerprint sensor directly on the card since at least 2004. If the privacy-preserving card is integrated within the cell-phone of the user, it is also possible to imagine that iris recognition could be easily implemented if the cell-phone possesses a camera.

which is public and will be used to check the authenticity of a privacy-preserving identity card during the group signature.

5.2 Mutual Authenticity Checking

Before the card will answer to questions of a particular reader, it needs to ensure that 1) the reader is an authentic device and 2) it possesses the corresponding credentials. On the other hand, the reader has to check that the card is a genuine privacy-preserving identity card but without learning any information related to the identity of the card or its user. Regarding the scheme used for signing the credential, any standard signature scheme such as DSA or ECDSA can be used to implement this functionality in practice. Efficient implementation of group signature with optional anonymity withdrawal exist such as the Camenish-Lysyanskaya signature scheme [8] which is proven secure in the random oracle model under the strong RSA assumption. The mutual authenticity checking protocol consists in three rounds of communication:

1. During the first round, the card generates dynamically for the session a pair of encryption/decryption keys (EK_{temp}, DK_{temp}). The public encryption key EK_{temp} will be used by the reader to secure the communication it sends to the card during this session while the decryption key DK_{temp} is kept secret. The card also computes $\sigma_{G,U}(EK_{temp}) \leftarrow \mathsf{SignGroup}(EK_{temp}, SKG_U)$, which corresponds to a group signature on the encryption key EK_{temp}. The card sends in clear EK_{temp} and $\sigma_{G,U}(EK_{temp})$ to the reader. The reader will consider the group signature as valid (and proceed to the second round) if $\mathsf{VerifySignGroup}(EK_{temp}, \sigma_{G,U}(EK_{temp}), VKG)$ outputs accept or aborts the protocol otherwise.
2. During the second round, the reader uses the card's public key to encrypt its credential cr, the signature of the CA on this credential $\sigma_{CA}(cr)$ as well as a randomly generated string of bits r, and sends this encrypted message to the card. The card performs $\mathsf{VerifySign}(cr, \sigma_{CA}(cr), VK_{CA})$ and either accept the reader and goes to the third round, or reject and aborts the protocol. The card should have a built-in mechanism that limits the number of attempts that a reader may try within some time window.
3. During the third round, the card computes $\sigma_{G,U}(r) \leftarrow \mathsf{SignGroup}(r, SKG_U)$, which corresponds to a group signature on the random string of bits r. Afterwards, the card sends to the reader the cipher $ciph \leftarrow \mathsf{Encrypt}(\sigma_{G,U}(r), EK)$, where $ciph$ corresponds to the encryption of the message $\sigma_{G,U}(r)$ with the public key EK. Finally, the reader decrypts this message by performing $\mathsf{Decrypt}(ciph, DK)$ which reveals $\sigma_{G,U}(r)$. If $\mathsf{VerifySignGroup}(r, \sigma_{G,U}(r), VKG)$ has for outcome accept, the reader recognizes the card has a genuine one. Otherwise, the reader aborts the protocol.

Suppose that the reader stores in a list all the pairs of random strings and group signatures $(r, \sigma_{G,U}(r))$ that he has seen along with other information such as a time stamp. As such this list is of no use for it to break the privacy of

users as it is not even able to recognize if two different signatures belong to the same individual or not. However in some extreme situation where there is a clear necessity of lifting the anonymity of a particular signature, the reader may hand over the pair $(r, \sigma_{G,U}(r))$ to the CA which will be able to retrieve SKG_U by performing $\mathsf{LiftAnonymity}(r, \sigma_G(r))$ and thus also the identity of U.

5.3 Biometric Verification

The privacy-preserving identity card is activated by the verification of the biometrics of its user. During this phase, a fresh biometric sample b' of the user is acquired by the biometric sensor and sends to the card which then performs the Verify operation upon it. This operation consists in computing $z \oplus b'$, decoding this towards the nearest codeword c' and calculating $h(c')$ to the card. The outcome of Verify is either accept or reject depending on whether or not $h(c') = h(c)$. If the user passes the verification test, the card is considered activated and enter the question-response protocol. Otherwise, the card refuses to answer to external communication.

5.4 Question-Response Protocol

Let $f(a_i)$ be the binary answer to a boolean question f about the attribute a_i of the user (or a combination of attributes). For instance, the semantic of the bit $f(a_i)$ could be true if its value is 1 and false if its value is 0. The question f as well as the public encryption key EK of the reader have been transmitted as part of the credential cr. First, the card concatenates the answer bit $f(a_i)$ with the random string r sent by the reader during the mutual authentication phase to obtain $f(a_i)||r$ and signs it, which generates $\sigma_{G,U}(f(a_i)||r)$. The card computes the cipher $ciph \leftarrow \mathsf{Encrypt}(f(a_i)||r||\sigma_{G,U}(f(a_i)||r), EK)$, where $ciph$ corresponds to the encryption of the message $f(a_i)||r||\sigma_{G,U}(f(a_i)||r)$ with the public key EK. Afterwards, the reader decrypts this message by performing $\mathsf{Decrypt}(ciph, DK)$ which reveals $f(a_i)||r$ and $\sigma_{G,U}(f(a_i)||r)$. The reader first verifies the validity of the signature $\sigma_{G,U}(f(a_i)||r)$ with the verification key of the group and believes the answer $f(a_i)$ only if this verification succeeds. Note that in the implementation BasicPIC , the correctness of answer $f(a_i)$ relies partly on the assumption that the card is tamperproof and therefore cannot be made to misbehave and lie to a question asked by the reader.

Consider an adversary that would like to play a relay attack by transmitting the communication normally between a genuine card and a genuine reader during the mutual authentication phase and then hijacks the session during the question-response protocol by acting as the card. If the answer bit was not signed with the private signature key of the card, the adversary could set the answer to the reader's question to his own choice.

The encryption scheme used has to be *semantically secure*[15] in order to avoid the possibility of an adversary having an advantage in guessing whether the an-

[15] Ideally, the encryption scheme should even fulfill a stronger security requirement called *indistinguishability under adaptive chosen ciphertext attack* (IND-CCA2)

swer of the card to the reader's question is 0 or 1. As a semantically secure encryption is necessary also probabilistic, this ensures that even if the card answers twice to the same question it will not be possible for an eavesdropper to distinguish whether these two answers where produced by the same privacy-preserving identity card or two different cards. In practice, this encryption scheme could be for instance the Cramer-Shoup cryptosystem [12] which has been one of the first proven to satisfy the IND-CAA2 property.

5.5 Analysis of the Implementation

In this Section, we will describe informally why the implementation BasicPIC fulfills the desiderata of a privacy-preserving identity card (as listed in Section 2) and analyze its cost. In details, the implementation BasicPIC respects the following properties:

- *No personal information leakage*: due its tamper-proof aspect, the attributes describing the user are safely stored on the smartcard and only one bit of information regarding the user is revealed every time the card answers a question.
- *Unlinkability*: the use of a group signature prevents the possibility of linking the proofs of two different statements to the same user. Moreover, there is no such thing as a pseudonym or an identifier used in our description of BasicPIC (with exception of the group signing key SKG_U, which is never disclosed by the card). In particular, there is no identity card number, which could be used to trace all the card uses and the public key for the session EK_{temp} is generated dynamically at random by the card and has no link with its identity.
- *Ownership proof*: before its activation, the card will check that the current user is effectively the legitimate owner of the card by verifying his biometrics.
- *Authenticity*: the reader will prove its authenticity and its right to ask a particular question by showing the corresponding credential signed by the CA. The card will prove its authenticity by showing that it can sign a randomly generated message on the behalf of the group of genuine privacy-preserving identity card.
- *Correctness*: in this implementation, the correctness of a statement proven by the card relies mainly on the fact that the tamper-proof properties of the smartcard forbids a dishonest user from changing its designed behaviour. Indeed, the card can be seen as a kind of oracle that never lies to a question asked to it. To change the behaviour of the oracle would require breaking the smartcard, which would violate the tamper-proof assumption. Moreover as the answer is encrypted using a non-malleable asymmetric encryption scheme using the public key of the reader, it is impossible for a potential adversary to flip the answer bit without being detected. Finally, as the answer bit is signed

(see [23] for instance). This property has been proven to also guarantee the *non-malleability* property and thus the threat of an adversary flipping the bit of the answer transmitted.

with the key of the card, this prevents an adversary from impersonating a valid card during the question-response protocol.
- *Unforgeability*: this property is ensured by the tamper-proofness of the smartcard, as well as by the verification of the credential issued by the CA.
- *Optional anonymity removing*: in extreme situations, the anonymity of the actions of the user of a privacy-preserving identity card can be lifted by having the CA cooperating with a verifier and applying the LiftAnonymity operation on the corresponding pair of random string and associated signature.
- *Explicit consent*: in most situations, the user expresses his consent by inserting his card in the reader: we can consider that, since the reader has to be certified by the CA, it is trustworthy enough, i.e., tamper proof and able to display correctly the question on a screen (part of the reader). Then if the user accepts the question, he just confirms it by pushing a switch, else he just withdraws his card from the reader. If the reader cannot be trusted and if the question can be too sensitive, the card should be equipped with embedded screen and switch.

Regarding practical considerations, the error-correcting codes used to construct the fuzzy commitment schemes generally admit efficient decoding procedures, which means that the decoding of $z \oplus b'$ towards the nearest codeword c' can also be done efficiently (even if the biometric sensor is directly integrated within the card). Efficient versions of group signatures for smartcards also exist and can be implemented with current technologies.

6 Extended Implementation

The main drawback of BasicPIC is that a great part of its security relies on the tamper-proof aspect of the smartcard. If this assumption is broken, for instance if the adversary is able to access the memory of the smartcard, this can greatly endanger security properties such as *no personal information leakage*, *authenticity*, *unforgeability* and *correctness*. To overcome this limitation, we propose in this section an extended implementation of the privacy-preserving identity card that we call ExtendedPIC. The main idea of this implementation is to complement the functionalities of BasicPIC with the use of *fuzzy extractors* to protect the information stored in the card and *non-interactive zero-knowledge proofs* as a privacy-preserving proof of statements related to the user's data.

6.1 Initialisation

The CA is responsible for signing the user's information in order to produce the anonymous credentials. The credentials emitted by the CA take the form of the CA's signature on the attributes of the user. Specifically, we denote these credentials by $\sigma_{CA}(a_1), \ldots, \sigma_{CA}(a_k)$, where $\sigma_{CA}(a_i)$ is the signature on the i^{th} attribute of the user using the CA secret key. The operation of the fuzzy extractor

Generate is performed on the biometric profile of the user b and produces as output a random string $rand$ and an helper string p. The random string $rand$ will be use as the key to encrypt[16] the attributes of the user, $a_1, \ldots, a_k$, and the signatures of the CA on these attributes $\sigma_{CA}(a_1), \ldots, \sigma_{CA}(a_k)$. The attributes and their associated signatures are stored encrypted inside the card but the helper string p can be stored unprotected.

Although it would be also possible to store an encrypted version of the signing key of the user SKG_U as well as z and $h(c)$, we suppose for the sake of clarity that it is not the case and that the mutual authenticity checking as well as the biometric verification are performed in the same manner as in BasicPIC (Sections 5.2 and 5.3). In practice however, SKG_U could also be encrypted using the key extracted from the fuzzy extractor, which requires that the biometric profile of the user is acquired first during the Retrieve operation in order for the mutual authenticity protocol to succeed. In this situation, it is possible to combine in a natural manner the biometric verification and the mutual authenticity checking into a single protocol. This protocol would fail if the biometric profile acquired during the Retrieve operation does not correspond to that of the valid owner of the card or if the card does not possess a valid private signature key SKG_U[17].

6.2 Privacy-preserving Proof of Statements

In our setting, the card wants to prove to the reader some function related to the attributes of the user and also that these attributes have been signed (certified) by the CA. However, we want to go beyond simply sending an answer bit, by issuing a zero-knowledge proof. This can be done as follows:

1. We suppose that the binary question asked by the reader is related to the i^{th} attribute of the user. The card performs Retrieve by taking as input a fresh biometric sample of the user b' and the helper string p stored on the card. The output of the Retrieve operation is the random string $rand$ which is used as a key to decrypt the values of the attribute a_i and its associated signature $\sigma_{CA}(a_i)$ from their encrypted versions stored on the card.
2. The card computes $comm(a_i) \leftarrow \mathsf{Commit}(a_i, aux)$, where $comm(a_i)$ is a commitment on the value of the i^{th} attribute a_i and aux is some auxiliary information needed to open the commitment. In practice, we propose to use the Groth-Sahai commitment scheme [17], which is perfectly binding (thus forbidding that the card can change afterwards the value of the attribute committed and therefore prove a false statement) and computationally hiding (thus preventing a reader to learn the value of the attribute committed unless he can break some computational assumption).

[16] For example, the encryption scheme used can be a symmetric scheme where $rand$ acts as the key for encrypting and decrypting data. For instance l, the size in bits of $rand$ can be set to be the size of an AES key (128 or 256 bits).

[17] An additional benefit of this protocol is to remove the need of using the fuzzy commitment scheme.

3. The card computes $\pi \leftarrow \mathsf{Prove}((a_i, \sigma_{CA}(a_i), aux) | \mathsf{VerifySign}(a_i, \sigma_{CA}(a_i), VK_{CA}) = \mathsf{accept} \wedge a_i = \mathsf{Open}(comm(a_i), aux) \wedge f(a_i) = true)$, where VK_{CA} is the public verification key of the CA that can be used to check the validity of the CA's signature, $\sigma(a_i)$ is the signature by the CA of attribute a_i and $f(a_i)$ is a boolean question regarding a_i. Effectively, π is a non-interactive zero-knowledge proof of the following statement "The user of this privacy-preserving identity card knows how to open the commitment *comm* to some value a_i, and this value has been signed by the CA, and when the boolean function f is computed on a_i it returns true" which could be summarized as "The CA certifies that the user of this privacy-preserving identity card satisfies the boolean question f when it is applied on his i^{th} attribute". The boolean question f could be any binary property related to an attribute of the user. The idea of using a zero-knowledge proof can also be extended so as to prove a binary statement regarding several attributes at the same time, such as a conjunction.
4. The card sends $\mathsf{Encrypt}(comm||\pi, EK)$ to the reader which then decrypts it and verifies the validity of the proof and outputs accept or reject.

For the practical implementation of the privacy-preserving proof of statements, we suggest to use the recent non-interactive zero-knowledge proofs developed by Belenkiy, Chase, Kohlweiss and Lysyanskaya [3]. These proofs are an extension of the CL-signatures [8] and have been proven secure on the common reference string model. These non-interactive zero-knowledge proofs are based partly on the Groth-Sahai commitment scheme [17] that has some interesting non-trivial properties such as being *f-extractable*, which means it is possible to prove that the committed value satisfies a certain property without revealing the value itself, and allows *randomizability*, which means that a fresh independent proof π' of the same statement related to the committed value can be issued from a previous proof π of this statement. In the context of the privacy-preserving identity card, the f-extractability property allows to show that an attribute of the user satisfies some binary property without disclosing the attribute itself whereas the randomizability property ensures that even if the card prove several times the same statement, the reader will see each time a different proof of this statement, thus avoiding the risk of linkability between them.

6.3 Analysis of the Implementation

The implementation ExtendedPIC fulfills the desiderata of a privacy-preserving identity card as it respects the following properties:

– *No personal information leakage*: the attributes of the user are stored in the smartcard encrypted and can only be decrypt if the user biometric profile is presented as input to the fuzzy extractor in conjunction with the helper string. Moreover, the card answers to a question of the card by showing a non-interactive zero knowledge proof which leaks nothing but one bit of information about the validity of a particular binary statement.

- *Unlinkability*: the use of a group signature prevents the possibility of linking the proofs of two different statements to the same user. Moreover, there is no such thing as a pseudonym or an identifier used in our description of ExtendedPIC (with exception of the group signing key SKG_U, which is never disclosed by the card). The randomizability property of the non-interactive zero-knowledge proof also ensures that even if the card proves several times the same statement, the proofs generated will be different and look as if they were independent.
- *Ownership proof*: before its activation, the card will check that the current user is effectively the legitimate owner of the card by verifying his biometrics. The biometric template of the user is also used as input to the fuzzy extractor when it is time to decrypt the data stored on the card during the privacy-preserving proof of statements.
- *Authenticity*: the reader will prove its authenticity and its right to ask a particular question by showing the corresponding credential signed by the CA. The card will prove its authenticity by showing that it can sign a randomly generated message on the behalf of the group of genuine privacy-preserving identity card, and also indirectly by showing the non-interactive zero-knowledge proof that it possesses the signature of CA on the attributes of the user.
- *Correctness*: the correctness of a statement proven by the card is a direct consequence of the soundness and completeness properties of the non-interactive zero-knowledge proof used. Moreover as the answer is encrypted using a non-malleable asymmetric encryption scheme using the public key of the reader, it is impossible for a potential adversary to flip the answer bit without being detected.
- *Unforgeability*: this property is ensured by the tamper-proofness of the smart-card, as well as the fact that the data of the user is stored encrypted on the card, plus by the verification of the credential issued by the CA and the signatures of the CA on the attributes of the user.
- *Optional anonymity removing* and *explicit consent*: these properties are ensured in the same manner than for the implementation BasicPIC (see Section 5.5 for more details).

7 Possible extensions and Conclusion

Potential applications of the privacy-preserving identity card may include access to online services such as e-government services. In this context, the card could be plugged to a standard personal computer via an external trusted USB reader certified and sold by the government. In this case, all the communication between the reader and the e-government platform hosting the online services should be encrypted to prevent potential information leakage, e.g. to a spyware that would have infected the user's personal computer. Of course, in this virtual context it may more difficult for a user to keep an explicit control on how his data are used and we may have to cope with more threats than in the simple card-reader interaction scenario.

A possible extension to the privacy-preserving identity card is to embed it directly in a device such as a cellular phone. Of course, this raises the question of how much trust can be put in such a device. Another extension is to use the privacy-preserving identity card as an electronic wallet by drawing on techniques such as one-use credentials. In such a scenario, the content of the card could be updated regularly via a terminal that is certified by the CA. For instance, when Alice buys a ticket concert from a vending machine she could upload the corresponding one-time credential on her privacy-preserving identity card.

Such extensions would require an in-depth security analysis to ensure that they can be safely integrated in a privacy-preserving identity card. But with the basic and extended implementations that we have described in Sections 5 and 6, it is technically feasible to develop and deploy a privacy-preserving identity card with currently available technologies. Whether governments and law enforcement authorities would accept such a card to be deployed is another question.

8 Acknowledgments

We would like to thanks the anonymous experts who reviewed a previous version of the paper for their insightful comments which have helped us improve the quality of this paper.

References

1. Bangerter, E., Camenisch, J. and Lysyanskaya, A., "A cryptographic framework for the controlled release of certified data", *Proceedings of the 12th International Security Protocols Workshop*, pp. 20–42, 2004.
2. Batina, L., Mentens, N. and Verbauwhede, I., "Side-channel issues for designing secure hardware implementations", *Proceeding of the 11th IEEE International On-Line Testing Symposium*, pp. 118–121, 2005.
3. Belenkiy, M., Chase, M., Kolhweiss, M. and Lysyanskaya, A., "P-signatures and noninteractive anonymous credentials", *Proceedings of the 5th Theory of Cryptography Conference (TCC'08)*, pp. 356–374, 2008.
4. D. Birch, "Psychic ID: A blueprint for a modern national identity scheme", *Identity in the Information Society* **1**(1), 2009.
5. Boudot, F., "Partial revelation of certified identity", *Proceedings of the First International Conference on Smart Card Research and Advanced Applications (CARDIS'00)*, pp. 257–272, 2000.
6. Calmels, B., Canard, S., Girault, M. and Sibert, H., "Low-cost cryptography for privacy in RFID systems", *Proceedings of the 7th International Conference on Smart Card Research and Advanced Applications (CARDIS'06)*, pp. 237–251, 2006.
7. Camenisch, J. and Lysyanska, A., "An efficient system for non-transferable anonymous credentials with optional anonymity revocation", *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'01)*, pp. 93–118, 2001.

8. Camenisch, J. and Lysyanskaya, A., "A signature scheme with efficient protocols", *Proceedings of the International Conference on Security in Communication Networks (SCN'02)*, pp. 268–289, 2002.
9. Camenisch, J. and Thomas, G., "Efficient attributes for anonymous credentials", *Proceedings of the 2008 ACM Conference on Computer and Communications Security (CCS'08)*, pp. 345–356, 2008.
10. D. Chaum, "Security without identification: transaction systems to make Big Brother obsolete", *Communications of the ACM* **28**(10), pp. 1030–1044, 1985.
11. Chaum, D. and van Heyst, E., "Group signatures", *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'91)*, pp. 257–265, 1991.
12. Cramer, R. and Shoup, V., "A public-key cryptosystem provably secure against adaptive chosen ciphertext attack", *Proceedings of the International Conference on Cryptology (CRYPTO'98)*, pp. 13–25, 1998.
13. Dodis, Y., Reyzin, L. and Smith, A., "Fuzzy extractors, a brief survey of results from 2004 to 2006", Chapter 5 of *Security with Noisy Data*. Tuyls, P., Skoric, B. and Kevenaar, T., editors. Springer-Verlag, 2007.
14. European Union, *Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995* on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Available at `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML`.
15. European Network and Information Security Agency (ENISA) position paper, "Privacy features of European eID card specifications". Available at `http://www.enisa.europa.eu/doc/pdf/deliverables/enisa_privacy_features_eID.pdf`.
16. Goldreich, O., Micali, S. and Wigderson, A., "Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems", *Journal of the ACM* **38**(3), pp. 691–729, 1991.
17. Groth, J. and Sahai, A., "Efficient non-interactive proof systems for bilinear groups", *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'08)*, pp. 415–432, 2008.
18. Juels, A. and Wattenberg, M., "A fuzzy commitment scheme", *Proceedings of the 6th ACM Conference on Computer and Communications Security (CCS'99)*, pp. 28–36, 1999.
19. Lysyanskaya, A., Rivest, R.L., Sahai, A. and Wolf, S., "Pseudonym systems", *Proceedings of the 6th Annual International Workshop on Selected Areas in Cryptography (SAC'99)*, pp. 184–199, 1999.
20. PRIME - Privacy and Identity Management for Europe, "PRIME white paper", May 2008, available at `https://www.prime-project.eu/prime_products/whitepaper/`.
21. Ravi, S., Raghuanathan, A. and Chadrakar, S., "Tamper resistance mechanisms for secure embedded systems", *Proceedings of the 17th International Conference on VLSI Design (VLSID'04)*, pp. 605–611, 2004.
22. Ratha, N., Connell, J. and Bolle, R., "Enhancing security and privacy in biometrics-based authentication systems", *IBM Systems Journal* **40**(3), pp. 614–634, 2001.
23. Shoup, V., "Why chosen ciphertext security matters", *IBM Research Report* RZ 3076, November 1998.