A Proposal for a Privacy-preserving National Identity Card

Yves Deswarte[1,2], **Sébastien Gambs**[3]

[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

[3] Université de Rennes 1 - INRIA / IRISA ; Campus Universitaire de Beaulieu, 35042 Rennes, France

E-mail: Yves.Deswarte@laas.fr, sgambs@irisa.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 him while minimizing personal information leakage. The privacy of the user is protected through the use of anonymous credentials that, 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 implementation proposals of the privacy-preserving identity card are described and discussed, and should be experimented on Java Cards in a near future. Possible extensions are also proposed as future work.

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*.

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.

[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, [20]).

[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.

[3] In this paper, we use the word "user" to denote at the same time both the owner and the effective user of the

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

Scenario (Alice in Anonymityland)**.** Alice is privacy-addicted since she has read the seminal paper of Chaum on anonymity [16]. 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 is 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 some of 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 boards a plane or to gain access to local services restricted to her neighborhood residents.

If Alice's card is lost or stolen, she does not need to worry about misuse for malicious purpose, thanks to 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, identical privacy-preserving identity card.

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, before describing some related work in Section 3. Afterwards in Section 4, 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 5, we briefly describe how such a card might be used in practice as well as our security assumptions, before in Section 6 and Section 7 proposing two practical implementations of the privacy-preserving identity card. Finally, we conclude in Section 8 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* [13] (see Section 4.2 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 a dedicated governmental agency) 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 the scenario an organization could be the immigration services, a theater or an airline company.) A *verifier* belongs to one organization and interacts with the

card. Indeed as the user needs to authenticate himself to the card before he can use it, the user of the card will effectively always be also his owner.

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.

Furthermore, we assume that the verifier is not allowed to ask arbitrary questions to a privacy-preserving identity card, but rather that he is entitled to ask *only one question*[4] directly related to the verifier's role and the resources to which it can grant access. The question that a particular verifier is allowed to ask, as well as his public encryption key, are specified in a credential signed by the CA. The public encryption key of the reader can be used by the card to communicate confidentially with the reader, whereas the decryption key is kept secret and is only known by the reader itself. The public verification key of the CA is embedded in each privacy-preserving identity card issued and can be used to check the validity of a certificate.

As illustrated in the scenario, ideally the privacy-preserving identity card system 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 be 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.

- *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*).

- *Non-transferability*: only the legitimate user should be able to use his privacy-preser–ving identity card to prove statements about himself to other entities. Otherwise, the user could be the victim of identity theft if he loses his card or even sell for some money the use of his privacy-preserving identity card to somebody else, thus deliberately transferring his privileges or even his identity to illegitimate users.

- *Authenticity and unforgeability*: it should be impossible to counterfeit the identity card of a user and to usurp his role, or to forge a fake card that corresponds to an arbitrary chosen identity. Moreover, it should be impossible for an adversary to impersonate the role of a valid privacy-preserving identity card or a valid reader.

Apart from these fundamental requirements, the privacy-preserving identity card may also satisfy additional properties that can be desirable in some cases, such as:

[4] The question can be a Boolean expression with AND, OR and NOT operators and with operands that are attributes of the owner, which can be verified according to data stored in the card.

[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.

- *Optional anonymity removing*: the actions of the user should stay anonymous[6] 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 (i.e. retrieving the identity of the owner of the card that has been used). 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 is indeed protected by the card.

- *Transparency and 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 to the user. Ideally, the privacy-preserving identity card should not be considered by the user as a black-box that magically protects his privacy, but rather he should have the possibility to view the history of interactions of the card with external readers. 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 Related Work

Our proposal for the privacy-preserving national identity card is close in spirit to the project PRIME[7] (PRivacy and Identity Management for Europe) [31], 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 a possible future U.K. national identity card [5], 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 a specific purpose if the user possesses the corresponding credential, and nothing otherwise. Before that Psychic ID accepts to answer a particular question, first the reader has to show a credential that attests of its right to ask this question. In a recent report [21], ENISA has analyzed the specifications of existing electronic identity cards in Europe with respect to some privacy requirements, and this analysis shows that no existing card fulfills all privacy requirements, in particular concerning unlinkability and minimal information disclosure.

Other related works close to our approach include a protocol for the partial revelation of information related to certified identity proposed by Boudot [8] and the development of a cryptographic framework for the controlled release of certified data due to Bangerter, Camenisch and Lysyanskaya [1].

In Boudot's scheme [8], a certified version of the personal information of a user is stored on a smartcard. This certification takes the form of a signature obtained from the CA. When

[6] This means that no information other than a binary statement should be disclosed by the use of the card. In some cases, such as when checking the validity of a boarding pass, the binary statement is a confirmation of the name and first name of the user (read from the boarding pass).

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

the user wants to prove some property related to its identity, he starts by sending a commitment of this information to the verifier and then issues a zero-knowledge proof that this information is certified *and* that it respects some property. For instance, it can be used to prove that the committed value lies within some interval (e.g. that Alice's age is between 18 and 25 years old). The protocol proposed by Boudot is based on the Fujisaki-Okamoto commitment scheme [23] and is secure under the strong RSA assumption. The privacy-preserving proof of statements (Section 7.2) that we used in our second proposal for implementation of the privacy-preserving national identity card was inspired from Boudot's seminal paper.

In the framework of Bangerter, Camenisch and Lysyanskaya [1], the user possesses a certificate containing his personal data that has been signed by the CA. When the user interacts with a verifier, he can choose which parts of the certificate he wants to disclose and also prove some property such that a particular value stored in the certificate lies within some interval (much like in Boudot's work). Contrary to traditional certificates, the proposed protocol allows for multiple show unlinkability as the user can prove several times the same statement to the same verifier without the possibility for the verifier to link these different proofs to the same entity. While under normal circumstances, the verifier only learns partial information relative to the user due to the selective disclosure, the proposed scheme also include a mechanism which reveals the identity of the user with respect to a particular interaction (i.e. lift his anonymity). This mechanism requires an active cooperation between the verifier and the authority who has issued the certificates. The framework also enables to prove relationships between different committed data.

4 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.

4.1 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[8] [2]. As in cryptology, there is an ongoing race in the smartcard world between the developers of attacks and the designers of countermeasures (see [32] 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[9], by

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

[9] This assertion is true at least for low-cost smartcards, even if public-key cryptosystems are available on most of recent smartcards, including Java Cards.

using algorithms such as triple DES (Data Encryption Standard) or CRYPTO-1[10]. 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 [12]. They have also described a low-cost version of a group signature scheme thus suggesting that such cryptographic primitive are possible even with inexpensive smartcard technologies. Even more recently, Bichsel, Camenisch, Groß and Shoup have conducted an empirical evaluation of the implementation of an anonymous credential system [4] (which is based on the Camenisch-Lysyanskaya signature scheme [14]) on JavaCards (standard 2.2.1). The implementation was using a RSA encryption with a modulus of 1536 bits and was autonomous in the sense that the smartcard was not delegating computations to the terminal or another possibly untrusted hardware. Their experiment was successful in showing that anonymous credential are within the reach of current technology as they obtained running time in the order of 10 seconds (which is several order of magnitude lower than previous implementations). Of course, for a user this time may still be too long and inconvenient for some practical applications but this is likely to improve in the future with new advances in the smartcard technology[11].

4.2 Anonymous Credentials

An *anonymous credential* is a cryptographic token that allows a user to prove statements about himself to verifiers anonymously. Anonymous credentials are generally based on *zero-knowledge* type of proofs [24] 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 credential has been proposed by Chaum [16] 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). Brands has designed efficient protocols for proving relations among committed values [10] which form the basis of the U-Prove anonymous credential system. However, these techniques do not display multiple-show unlinkability. 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 [17] have been introduced by Chaum and van Heyst to provide anonymity to the signer of a 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 signa-

[10] CRYPTO-1 is a proprietary encryption algorithm from NXP Semiconductors used mainly for applications in contactless cards, such as the MIFARE RFID tags. The security of this encryption scheme has been almost completely broken by a serie of recent attacks highlighted in the security community.

[11] For instance, Bichsel, Camenisch, Groß and Shoup have used a Java Card with a 3.57 MHz 8-bits processor [4], while some smartcards exist with 32-bits processors running at 66 MHz (e.g., the Infineon SLE 88CF4000P).

ture 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 $\mathsf{SignGroup}$ 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 $\mathsf{VerifySignGroup}$ 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 $\sigma_{G,U}(m)$. $\mathsf{VerifySignGroup}$ 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 $\mathsf{LiftAnonymity}$ 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 identifying the private signature key SKG_U from the signature and then retrieving the identity associated to this key.

The Identity Mixer (Idemix) project from IBM and the Direct Anonymous Attestation (DAA) protocol [11] adopted for the anonymous authentification of Trusted Computing Platform (TPM) are two famous examples of anonymous credentials based on the concept of group signature.

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 the knowledge of 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 using 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.

4.3 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 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 into a vector of 512 bytes called the IrisCode. Afterwards, it is fairly simple to evaluate the dissimilarity between two codewords just by computing the Hamming distance between these two vectors.

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* [33] 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* [28] that we summarize here. For the sake of clarity, we assume that b, the biometric profile of the user, can be represented as a binary vector of length n (i.e., $b \in \{0,1\}^n$)[12]. 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 FAR and FRR. A hash function h is also used by the protocol. The fuzzy commitment scheme [28] 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[13] measures a fresh biometric sample b' and sends it to the card at the beginning of the Verify operation. Afterwards, the card computes $z \oplus b'$

[12] 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 the particular situation.

[13] Such a sensor can be implemented on the smartcard itself, but in practice it may also be part of the smartcard reader device.

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 [19]). This primitive allows to extract a uniformly distributed random string $rand \in \{0,1\}^l$ [14] 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. The dissimilarity measure $dist$ can be for instance the Hamming distance, the set difference or the edit distance [19]. When initialized for the first time, a fuzzy extractor outputs a helper string called $p \in \{0,1\}^*$, which will be part 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. Regarding the practical applicability of these technologies, Hao, Anderson and Daugman [26] have demonstrated that by relying on Iris code for biometric authentication, it is possible to retrieve up to 140 random bits of key (more than needed for a 128 bits AES key), while displaying very low rates of false acceptance (0%) and false rejection (0.47%).

5 Operation and Use of the Privacy-preserving Identity Card

Security assumptions. We assume that the privacy-preserving identity card is a contact smartcard that has sufficient resistance against physical and logical attacks (see Section 4.1). This requirement of tamper-resistance is essential for the security of our first implementation proposal for the privacy-preserving identity card, BasicPIC (see Section 6). This requirement can be relaxed by using fuzzy extractors for our extended proposal, ExtendedPIC (see Section 7). We also assume that the smartcard reader device that will interact with the privacy-preserving identity card possesses similar tamper-proof properties. However, if this assumption fails for the reader, the consequences may not be as disastrous as for the identity card. For instance, if an adversary can break the tamper-resistance of a reader, this mainly means he can have access to its private decryption key and its certificate. This

[14] 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.

would give enough information for an adversary to produce clones of the reader (thus impersonating a genuine reader) but as such cannot be used to forge a fake privacy-preserving identity card or to obtain more information from the card than the genuine reader.

The smartcard contains a processor that can compute efficiently cryptographic primitives such as (pseudo-) random number generation[15], asymmetric encryption and group signature scheme. The reader is also assumed to have at least the same cryptographic abilities. 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. We assume that the acquisition during the biometric authentication is done via a trusted sensor that is either directly integrated on the card or on a remote terminal. In this latter case besides the tamper-resistance assumption on the sensor, we also need to assume the availability of a secure channel between the biometric sensor and the card. With regard to practical issues, we recommend to rely either on a strong biometric such as the iris recognition, which is quite difficult to counterfeit, or the combination of a weak biometric such as fingerprint[16] with a PIN. However, this adds another assumption than the keyboard needed for entering the PIN needs also to be trusted and than the communication between this keyboard and the card needs to be done through a secure channel (unless the keyboard is directly integrated on the card).

High-level view of the protocol and use of the card. When the smartcard is inserted into a reader device, the smartcard processor initiates a *mutual authentication* between the card and the reader (Section 6.2) for the reader to be sure that the card has been issued by a CA and for the card to be sure that the reader has a genuine certificate. 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 (Section 6.3), by using for instance the fuzzy commitment scheme for biometric authentication described in Section 4.3. Finally, when the biometric authentication is successful, the processor initiates a *question-response* protocol with the reader device (Section 6.4).

In practice, the question asked by 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 be a Boolean expression on several attributes of the user (for instance "Is the user under 18 years old OR over 60 years old AND an inhabitant of Toulouse?").

[15] Note that a hardware-based random number generator is integrated in most of the current Java Cards (e.g., Oberthur ID-One Cosmo 32 v5, NXP JCOP v2.2/41, SHC1206, . . .).

[16] We called a fingerprint a "weak biometric" because it can be collected and copied much more easily (for instance through simple "wax finger" techniques) than a "strong biometric" such as the iris for which the recognition process may involved the observation of the behaviour of the iris under chosen lighting conditions. Note also that it is possible to mitigate the threat of biometric cloning by using mechanisms checking the *liveness condition*.

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. This is similar to a recent work by Camenisch and Thomas [15], which provides an efficient implementation of anonymous credentials that allows to prove AND, OR and NOT statements regarding the attributes encoded. Even in the situation where the reader is allowed to ask a complex binary question related to several attributes of the user (as specified in the reader's certificate issued by the CA), the reader only learns one bit of information through this interaction with the card. Of course, this would not be case if the reader was allowed instead to ask several elementary binary queries and compute itself the Boolean expression.

Note that the role of the card is mainly to certify a binary statement related to personal data stored in the card, corresponding to a question that the reader is authorized to ask. For that:

- The reader must already know the values of the attributes (or their ranges) concerned by its question.
- These values are either constant, built-in the reader (e.g., under 18 or over 60) or already available in the real world (e.g., the name of the passenger written on a boarding pass).
- In all cases, the disclosed personal information is minimal and it corresponds to the information necessary for the reader to performs the check it is designed and authorized to do.

There is however an inherent limitation on how far the unlinkability property can be pushed in a world where several organizations starts to record all the interactions they had with users through anonymous credentials and in which contexts these interactions occured, and pool this data together [22, 30]. Consider for instance, the scenario where a teenager has to prove to a theater that he is less than 18 years old in order to get a discount and then that 2 hours later (right after the end of the movie), he proved to the nearby swimming pool that he is less than 18 years old and he is an inhabitant of Toulouse (where the swimming pool is located) to get the lowest price for the entry ticket. In this scenario, even if the teenager proves the two statements related to his identity in a theoretically anonymous and unlinkable way, it may happen if the theater and the swimming pool collude that they are able to infer that the identity behind the two statements has a non-negligible probability to be the same individual.

6 Basic Implementation

The first implementation of the privacy-preserving identity card that we proposed combines the different technologies and concepts briefly reviewed in Section 4. We call it BasicPIC, which stands for Basic implementation of a Privacy-preserving Identity Card (PIC). In this implementation, we suppose that the smartcard tamper-proofness is "sufficient". In practice however, it is quite likely that if an adversary spends enough resources and time, he will be able to break the tamper proof characteristics of the smartcard and then 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.

We note that our BasicPIC proposal is similar in spirit to other approaches to biometrics-based non-transferable anonymous credentials, such as the seminal paper by Bleumer [7] and subsequent work by Implagliazzo and More [27]. The main idea behind all these approaches is to combine the use of biometric authentication with physical security by means of a tamper-resistant token responsible for storing the credentials. If the biometric authentication is not successful, then the physical token refuses to use the credentials it possesses, thus ensuring the property of non-transferability.

6.1 Initialisation

When the user wishes 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 request. We denote by $a_1, \ldots, a_k$, the k attributes of the user that embody 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 and other claimed attributes of the user, the RA scans the user's biometric profile 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 4.2) 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 as cleartext 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[17] 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[18]. 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 which

[17] Some companies, such as Novacard, have started to sell smartcards 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.

[18] The reader should also be tamperproof enough to protect its secret decryption key DK_R, thus preventing an attacker to forge cloned readers. In any case, a forged reader would not be able to retrieve more personal information from the BasicPIC than a genuine reader.

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

6.2 Mutual Authenticity Checking

Before the card answers the 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 Camenisch-Lysyanskaya signature scheme [14] which is proven secure in the random oracle model under the strong RSA assumption. The mutual authenticity checking scheme we propose is inspired by the "simplified TLS key transport protocol" presented in [9]. It consists of 4 rounds of communication:

1. During the first round, the card generates a random string of bits r_1 and sends it in clear to the reader.

2. During the second round, the reader sends in clear to the card a randomly generated string of bits r_2, its credential cr as well as the signature of the CA on this credential $\sigma_{CA}(cr)$. The card then checks if the function $\mathsf{VerifySign}(cr, \sigma_{CA}(cr), VK_{CA})$ returns accept, in which case the card goes to the third round, or reject and the card 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 so as to counter brute force attacks.

3. At the beginning of the third round, the card generates dynamically a random string PMK (called the pre-master secret) and computes from it $K_{CR} = MAC_{PMK}(r_1||r_2)$ using a message authentication code (MAC) in which PMK plays the role of the secret key. K_{CR} will serve as a session key for a symmetric cryptosystem (such as AES). It needs to be recomputed also by the reader, and thus PMK must be transmitted by the card to the reader, encrypted by the reader's public key EK_R (retrieved by the card from the readers credential cr): $ciph1 \leftarrow \mathsf{Encrypt}(PMK, EK_R)$. The card also computes $\sigma_{G,U}(h(r_1||r_2||ciph1))$, which corresponds to a group signature on the hash of the message composed of the concatenation of r_1, r_2 and $ciph1$. The card then sends to the reader $ciph1$ and $\sigma_{G,U}(h(r_1||r_2||ciph1))$. If $\mathsf{VerifySignGroup}(h(r_1||r_2||ciph1), \sigma_{G,U}$ $(h(r_1||r_2||ciph1)), VKG)$ has for outcome reject, the reader aborts the protocol. Otherwise, the reader recognizes the card has a genuine one. Afterwards, the reader decrypts the first part of the message by performing $\mathsf{Decrypt}(ciph1, DK_R)$ which reveals PMK and from it the reader can deduce $K_{CR} = MAC_{PMK}(r_1||r_2)$.

4. During the fourth round, the reader computes the encryption $ciph2 \leftarrow \mathsf{EncryptSym}$ $(h(r_1||r_2||ciph1||\sigma_{G,U}(h(r_1||r_2||ciph1)), K_{CR})))$, which is the encryption under a symmetric cryptosystem of the message $h(r_1||r_2||ciph1||\sigma_{G,U}(h(r_1||r_2||ciph1))$ (i.e. the hash of all the messages exchanged so far with the exception of its credential) under the key K_{CR} and sends it to the card. Finally, the card can decrypt it using the shared key K_{CR} and verifies the consistency of $h(r_1||r_2||ciph1||\sigma_{G,U}(h(r_1||r_2||ciph1))$ will all the messages exchanged so far. If this verification fails, the card aborts the protocols whereas otherwise the mutual authenticity checking is considered successful.

Suppose that the reader stores in a list all the pairs of message sequences and group signatures $(h(r_1||r_2||ciph1), \sigma_{G,U}(h(r_1||r_2||ciph1)))$ 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 $(h(r_1||r_2||ciph1), \sigma_{G,U}(h(r_1||r_2||ciph1)))$ to the CA which will be able to retrieve SKG_U by performing LiftAnonymity$(h(r_1||r_2||ciph1), \sigma_{G,U}(h(r_1||r_2||ciph1)))$ and thus also the identity of U.

6.3 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 sent 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.

6.4 Question-Response Protocol

Let $f(a_i)$ be the binary answer to a Boolean question f about the user's attribute a_i (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_R of the reader have been transmitted as part of the credential cr. First, the card concatenates the answer bit $f(a_i)$ with the common secret shared with the reader K_{CR} that was generated during the mutual authentication phase to obtain $f(a_i)||K_{CR}$ and signs it, which generates $\sigma_{G,U}(f(a_i)||K_{CR})$. The card computes the cipher $ciph \leftarrow$ Encrypt$(f(a_i)||K_{CR}||\sigma_{G,U}(f(a_i)||K_{CR}), EK_R)$, where $ciph$ corresponds to the encryption of the message $f(a_i)||K_{CR}||\sigma_{G,U}(f(a_i)||K_{CR})$ with the public key EK_R. Afterwards, the reader decrypts this message by performing Decrypt$(ciph, DK_R)$ which reveals $f(a_i)||K_{CR}$ and $\sigma_{G,U}(f(a_i)||K_{CR})$. The reader verifies the validity of the signature $\sigma_{G,U}(f(a_i)||K_{CR})$ 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.

The encryption scheme used has to be *semantically secure*[19] in order to avoid the possibility of an adversary having an advantage in guessing whether the answer of the card to the reader's question is 0 or 1. As a semantically secure encryption is necessarily 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 the above protocol, we

[19] Ideally, the encryption scheme should even fulfill a stronger security requirement called *indistinguishability under adaptive chosen ciphertext attack* (IND-CCA2) (see [34] for instance). This property has been proven to also guarantee the *non-malleability* property and thus counters the threat of an adversary flipping the bit of the answer transmitted.

have adopted the Cramer-Shoup cryptosystem [18] which has been one of the first proven to satisfy the IND-CAA2 property.

6.5 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 uses of the card. The common shared key K_{CR} is generated dynamically at random during the session and has no link with the identity of the 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 or the attributes values. 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 with the key of the card, this prevents an adversary from impersonating a valid card during the question-response protocol.

- *Non-transferability*: before entering the question-response protocol, the card will check that the current user is effectively the legitimate owner of the card by verifying his biometrics.

- *Authenticity and unforgeability*: 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 cards. Moreover, the card unforgeability is ensured by the tamper-proofness of the smartcard and the validity of the group signing key assigned by the CA, whereas he reader's unforgeability is guaranteed its tamper-proofness and by the validity of the reader's 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 message sequences and associated group signature.

- *Transparency and 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., tamperproof 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.

Note that one of the biggest threat against the protocol would be a *man-in-the-middle attack* where the adversary would sit between a genuine card and genuine reader. For instance, the adversary might want to prove to a reader a boolean statement that in reality is false (according to the data stored on the card) or simply break the privacy of the bit answered by the card. The purpose of the session key is mainly to prevent this kind of attack by ensuring the authenticity property during the mutual authenticity checking phase (Section 6.2) and during the question-response protocol (Section 6.4).

Moreover, depending on the biometric feature used for the user's authentication, it might be more or less difficult for an adversary to acquire a biometric sample of the user and to create a synthetic prosthesis of it that could be used to pass the biometric verification. For instance, a fingerprint is much easier to capture (for instance by stealing a glass that was used by an individual) than the behaviour of an iris under varying lighting conditions. In the latter case, it is much harder to generate a prosthesis of an artificial iris than that of a fingerprint. In all cases, if possible with respect to the technology of the card, the ideal solution would be to integrate several authentication methods into the card to obtain a stronger authentication mechanism (for instance combining biometric verification + PIN).

Regarding practical considerations, the smartcard used for the implementation of BasicPIC is required to have some cryptographic capacities such as (1) a (pseudo-)random number generator, (2) a cryptographic hashing function, (3) a generator for session key pairs, (4) a semantically secure encryption function, (5) a public-key signature verification function, (6) a group signature function and (7) an error-correcting code with efficient decoding procedures. Current JavaCards already integrate built-in libraries which contain optimized code for performing requirements (1) to (5) such as for instance SHA-1 for hashing and DSA for the signature scheme. Efficient versions of group signatures for smartcards, such as Camenisch-Lysyanskaya [14], also exist and can be implemented with current technologies as recently shown by Bichsel, Camenisch, Groß and Shoup [4]. Moreover, 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). In terms of memory, BasicPIC is quite efficient as it requires to be able to store the k attributes of the user (space complexity of $O(k)$), the private signature key of the card as well as the public verification key of the CA, z whose size is directly proportional to the biometric template and $h(c)$ which is of constant size.

7 Extended Implementation

The main drawback of BasicPIC is that a great part of its security relies on the tamperproof 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 some 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. More precisely, the properties of authenticity, unforgeability and non-transferability rely on the fuzzy extractors while the properties no personal information leakage, unlinkability and correctness are now provided by the non-interactive zero-knowledge proofs.

We note that recently Blanton and Hudelson have proposed independently an approach similar to ours in which the tamper-proofness assumption is also removed by using fuzzy extractors [6]. In particular even if the integrity of the device holding the credentials is breached, it is still impossible to recover either the biometric data of the user or his credentials. The only assumptions that we need to make are that the biometric acquisition is done through a trusted sensor, which will erase the biometric data it has captured once the biometric authentication has been completed and that the communication between the sensor and the card is done through a secure channel. Of course, these assumptions may be relaxed if the biometric sensor is integrated in the card.

7.1 Initialisation and Authentication

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 a_i, the i^{th} user attribute, generated by using the CA's 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 used as the key to encrypt[20] 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 in cleartext.

For the sake of simplicity, we can consider that the user signing key SKG_U as well as z and $h(c)$ are stored in cleartext and that the mutual authenticity checking as well as the biometric verification are performed in the same manner as in BasicPIC (Sections 6.2 and 6.3). In practice however, SKG_U should 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. At the beginning of this protocol, a fresh template of the current user of the card would be taken and fed as input to the Retrieve operation along with the helper string p. This would generate an random string $rand'$ which would be used as a secret key to decrypt the data stored on the card before launching the mutual authentication protocol which then proceeds as in the BasicPIC implementation. 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 (because the private signature key retrieved would not be valid) or if the card does not possess a valid private signature key SKG_U, which is a form of combination of the biometric verification together

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

with the mutual authentication[21].

7.2 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 [25], 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).

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. Moreover, by negating $\neg(f(a_i) = true)$ in the expression of the zero-knowledge proof, it is possible to prove that $f(a_i) = false$ (i.e., that the Boolean function f returns false when it is applied on his i^{th} attribute). The idea of using a zero-knowledge proof can also be extended so as to prove a Boolean expression regarding several attributes at the same time.

4. The card sends $\mathsf{Encrypt}(comm||\pi, EK_R)$ 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 [14] 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 [25] that

[21] In this protocol, the fuzzy commitment scheme is no more used, being replaced instead by the fuzzy extractor.

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.

7.3 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 decrypted if the user biometric profile is presented as input to the fuzzy extractor in conjunction with the helper string. Moreover, the card answers a question asked by the reader 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.

- *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.

- *Non-transferability*: 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 an input to the fuzzy extractor when it is time to decrypt the data stored on the card during the privacy-preserving proof of statements. As a consequence, without the presentation of a valid biometric sample of the user it is impossible to unlock the credentials stored on the card.

- *Authenticity and unforgeability*: 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 cards, and also indirectly by showing the non-interactive zero-knowledge proof that it possesses the signature of CA on the attributes of the user. The unforgeability is ensured by the tamper-proofness of the smartcard, 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, transparency and explicit consent*: these properties are ensured in the same manner than for the implementation BasicPIC (see Section 5.5 for more details).

The ExtendedPIC implementation is clearly more demanding in terms of ressources than the BasicPIC one, the most costly part being the non-interactive zero knowledge proof [3].

Note that in the implementation in which the biometric verication and the mutual authenticity checking are combined into a single protocol, a sample has to be captured by the biometric sensor and processed by the card before the mutual authenticity checking between the card and reader, as the data needed to conduct this step is encrypted by using a key derived from the user's biometrics. This is fine if the biometric sensor is integrated directly within the card but might be dangerous if the sensor is part of the reader itself when this reader is non-genuine. In this situation, the acquisition of the biometric sample of the user allows to decrypt the data stored in the card, which corresponds to an attack on the confidentiality of this data. However, this is possible only if the adversary can steal the card and successfully perform a physical violation of the hardware protection of the card and read out the encrypted data, which is not supposed to be easy. Moreover, the impact of this attack is mainly restricted to the privacy of the data stored on this particular card as it does not enable an adversary controlling the fake reader to forge a card corresponding to the identity of his choice. More specifically, once he has read this data the adversary might be able to produce clones of this card but not to forge an arbitrary card. By drawing on the revocation mechanism of group signature, it would also be possible to integrate the ability of revoking a particular PIC into the architecture of the system. For instance, if a card has been stolen or detected as being compromised, its private key of the group signature could be revoked and placed in a "blacklist" to make genuine readers reject future use of the card. However, this require the possibility of regularly updating this blacklist inside the reader's memory, which might be impractical for some applications.

8 Possible Extensions and Conclusion

Originally, the main purpose of the PIC is to enable a person to conduct tasks in the real world without having to disclose his identity, but the same device could potentially be used to access to online services such as e-government services or even e-business applications. We list here some possible extensions to the original concept of PIC:

- *Access to e-government services*. In case of online access to e-government services, the card could be plugged to a standard personal computer via an external trusted USB reader certified by the government. The e-government services could include for instance the online declaration of income, the consultation of the user's file as recorded in the database of a certain ministry or printing some official documents related to the identity of the user. 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. Of course, this does not preclude the risk of a spyware infecting the user's personal computer and gathering personal information as the user interacts with software and hardware installed on his machine during his access to e-government services. This is a serious threat that should be dealt with by using common security techniques such as antivirus and malware detection tools.

- *Access to e-business applications.* Another online extension would be to use the PIC as an authentication mean during access to e-business applications. For instance, when paying a purchase on Internet, the PIC could be used to show that the ownership of the credit card used for the payment is verified through the PIC. Using a system of anonymous credentials, it would be possible for the user to prove this statement anonymously (i.e., without having to disclose explicitly his identity to the e-business server). Of course, in this virtual context it may be 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 presented in the previous section. For instance, we could imagine some kind of phishing attack where the user receives an email making some advertisement for a particular online store together with an associated website address. If he is the malicious owner of this website, a malicious adversary could sit on the link between the card and the server and performs a man-in-the-middle attack. More precisely, the adversary would pretend to be a genuine online store to the card and relay the answers provided by the card to the real store and vice versa and thus gain access to resources in the behalf of the owner of the PIC (imagine for instance that the adversary makes the user of the PIC pay for the tunes he downloaded from an online music store). Note that the same kind of attack may also apply for the access to e-government services in which case it could lead to a privacy breach where the adversary learns personal information related to the user of the PIC (which could be used later for fraudulent ends such as identity theft). To prevent such attacks, a secure end-to-end channel could be established between the server and the trusted USB card reader (with mutual authentication of the server and the reader), and this even before the user-reader mutual authentication occurs. Afterwards, the same secure channel should be used for all communications exchanged between the user and the server during the session.

- *E-voting*. Consider also the scenario where the card is used for authentication during an election where it is possible to vote electronically. One can imagine a protocol where the user can prove his right to vote in a privacy-preserving manner using the PIC and such that he can cast only one vote. For instance, this protocol could rely on a one-show anonymous credential (instead of a multiple show) to provide anonymity for the voter while ensuring that he can vote only once. However, even if this application protects the identity of the voter and his choice, it does not address the usual threat of coercion, which is a recurrent problem of the e-voting setting. Someone could for instance point a gun at you when you are voting from your home or buy your vote and ask to be there as a witness when you cast your vote online.

- *Integration within a cell phone.* Another 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. Indeed, the user has to trust that his cell phone will only access his PIC in a rightful way and will not try to collect information about him (e.g., by recording the questions and their answers). Normally the answers of the PIC are encrypted at the beginning of the chain (in the PIC itself) and not accessible to the phone, however in some situations it is possible to deduce them indirectly, for instance if the user gain access to a resource such as a file after his interaction with an online service. Moreover, simply registering the questions can be used to trace the actions of the user (for instance if he often tries to have a discount when going to his neighborhood swimming pool) and thus constitute a form of profiling. To trust that a cell-phone is perfectly secure may be more demanding than trusting that a PIC and a reader (which have been certified by the government or an independent authority) are genuine. One of the advantage of using a cell-phone is that it can provide the contactless facilities but without the risk of the usual contactless smartcards which can be skimmed without even the user noticing it. Another advantage is to extend the biometric diversity available, for instance by using the integrated phone camera for iris recognition or its microphone for voice recognition. The phone screen can also be used to display various information (including the question asked) and its keyboard can be used for obtaining the consent of the user with an active confirmation. The PIC may also use the computational and memory capacities of the phone to its own benefit. For instance, the phone may add another level of encryption to the output of the PIC or register questions asked so far to the PIC and refuse to pass a question to the card when it decides that this question may endanger the privacy of the user (which is a form of profiling, but this time for the benefit of the user's privacy).

- *Use as an electronic wallet*. Another extension is to use the PIC as an electronic wallet by drawing on techniques such as one-time anonymous credentials. In such a scenario, the content of the card could be updated regularly via a certified terminal. The one-time credentials can play the role of electronic cash but also be used as e-ticket. For instance, when Alice buys a concert ticket from a vending machine she could upload the corresponding one-time credential on her privacy-preserving identity card while paying with anonymous e-cash. Later, the one-time credential of her ticket could be transmitted or displayed at the entrance of the concert hall via Alice's cellphone. Another application may include buying electronic tokens to access a transport system.

- *Integration of biometric sensor, display screen and keyboard within the card*. If the card integrates directly a biometric sensor, a display screen and a keyboard (or at least a single key), this can greatly enhance the trust of the user regarding the PIC as he does not need to trust anymore another apparatus such as an external biometric sensor or the screen of the reader or of his cell phone. Indeed, if embedded within the PIC, the biometric sensor can communicate directly with the smartcard thus limiting the risk of someone eavesdropping the communication channel. Moreover, it makes it more difficult for an adversary to acquire the biometric data of the user as the biometric sensor is directly in the hands of the user himself. The screen can be used to display the questions asked to the card but also its answers. Suppose for instance that the current user needs to prove that he is the owner of a particular PIC and that when he puts his fingerprints on the biometric reader his face is displayed on the screen of the card. This does not give any new information except that the current user is also the owner of the card (unless the card has been tampered with).

Such extensions would require an in-depth security analysis to ensure that they can be safely integrated in a privacy-preserving identity card but it is technically feasible to develop and deploy such an extended privacy-preserving identity card with currently available technologies. One fundamental question is whether or not developing a identity card achieving many functionalities differing from its original purpose (as it is currently the trend in many countries) is really a good idea at all. Indeed, it constitutes a *single point of failure* that if compromised will have serious consequences including a major impact on the privacy of the user.

Acknowledgements

The authors thank the anonymous reviewers for their constructive remarks that helped in improving 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] Bichsel, P., Camenisch, J., Groß, T. and Shoup, V., "Anonymous credentials on a standard java card", *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09)*, pp. 600–610, 2009.

[5] D. Birch, "Psychic ID: A blueprint for a modern national identity scheme", *Identity in the Information Society* **1**(1), 2009.

[6] Blanton, M. and Hudelson, W., "Biometric-based non-transferable anonymous credentials", *Proceeding of the 11th International Conference on Information and Communication Security(ICICS'09)*, pp. 165–180, 2009.

[7] G. Bleumer, "Biometric yet privacy protecting person authentication", *Proceedings of the 2nd International Workshop on Information Hiding*, pp. 99–110, 1998.

[8] F. Boudot, "Partial revelation of certified identity", *Proceedings of the First International Conference on Smart Card Research and Advanced Applications (CARDIS'00)*, pp. 257–272, 2000.

[9] Boyd, C. and A. Mathuria, *Protocols for Authentication and Key Establishment*, Springer-Verlag, 2003.

[10] Brands, S., *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*, MIT Press, 2000.

[11] Brickell, E., Camenisch, J. and Chen., L. "Direct anonymous attestation", *Proceedings of the 11th of the ACM Conference on Computer and Communications Security (CCS'04)*, pp. 225–234, 2004.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] D. Chaum, "Security without identification: transaction systems to make Big Brother obsolete", *Communications of the ACM* **28**(10), pp. 1030–1044, 1985.

[17] 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.

[18] 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.

[19] 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.

[20] 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`.

[21] 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`.

[22] Franz, M., Pashalidis, A. and Meyer, B., "Attacking unlinkability: the importance of context", *Proceedings of the 7th Workshop on Privacy Enhancing Technologies (PET'07)*, pp. 1–16, 2007.

[23] Fujisaki, E. and Okamoto, T., "Statistical zero-knowledge protocols to prove modular polynomial relations", *Proceedings of the International Conference on Cryptology (CRYPTO'97)*, pp. 16–30, 1997.

[24] 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.

[25] 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.

[26] Hao, F., Anderson, R. and Daugman, J., "Combining cryptography with biometrics effectively", *IEEE Transactions on Computers* **55**(9), pp. 1081–1088, 2006.

[27] Impagliazzo, R. and Miner More, S., "Anonymous credentials with biometrically-enforced non-transferability", *Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society (WPES'03)*, pp. 60–71, 2003.

[28] 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.

[29] 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.

[30] Pashalidis, A. and Meyer, B., "Linking anonymous transactions: the consistent view attack", *Proceedings of the 6th Workshop on Privacy Enhancing Technologies (PET'06)*, pp. 384–392, 2006.

[31] PRIME - Privacy and Identity Management for Europe, "PRIME white paper", May 2008, available at `https://www.prime-project.eu/prime_products/whitepaper/`.

[32] 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.

[33] 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.

[34] Shoup, V., "Why chosen ciphertext security matters", *IBM Research Report* RZ 3076, November 1998.