University of Twente
Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS)

Master Thesis

Secure & privacy-preserving eID systems with Attribute-based credentials

Brinda Badarinath Hampiholi
s1480197
brinda2089@gmail.com

Graduation Committee:
Dr. F.E Kargl (UT)
Dr. Andreas Peter (UT)
Hans de Jong (NXP)

UNIVERSITEIT TWENTE.

Abstract

National electronic identification (eID) systems aim to provide universal, unique and reliable identification and authentication mechanisms to the citizens. Many countries in Europe have already introduced or are about to introduce electronic ID cards to their citizens. The increasing number of eID infrastructures and initiatives have been taken to scale the eID systems to support both eGovernment and eCommerce services and this has necessitated security and privacy to be of highest order in all the use-case scenarios. As the ID documents are the carriers of a citizen's identity and personal data, the outgoing information from the cards must be restricted in order to prevent unwanted data disclosure, subsequent data misuse and loss of the citizen's privacy. The German eID system (nPA) is by-far the most advanced and privacy-preserving eID solution that has been launched in Europe as it has taken extra measures to protect its citizen's data and privacy; however some security and privacy threats persist in nPA.

In this thesis, we study the key concepts and security protocols used in nPA and identify its main threats. Furthermore, we explore Attribute-Based Credentials (ABC) by studying in detail about I Reveal My Attributes (IRMA), that is a recent technology built upon the principles on ABC. IRMA is a partial implementation of Idemix specification and it mainly focusses on enhancing security and privacy of identity management systems. IRMA makes use of zero-knowledge protocols to prove the validity of certain attributes of the eID cardholder while allowing the selective disclosure of attributes and supporting unlinkability features. We elaborate on the advantages of such ABCs that could be used to address the threats identified in nPA and propose a specific scheme that integrates IRMA authentication with nPA. Our proposal shows how to use IRMA in eID systems like nPA in order to overcome the its prevalent shortcomings. The IRMA authentication provides better privacy, security and flexibility for the eID infrastructure. Finally, we discuss the performance of smart-card implementation of IRMA credentials and present some use-cases that would benefit from the proposed eID scheme.

Keywords: eID systems, nPA, IRMA, authentication, privacy, user-control, attribute-based credentials, data minimization, unlinkability

Acknowledgements

Being a masters student in the EIT ICT Labs Master School, I conclude my second year of Master of Science, Security & Privacy course at University of Twente (UT), The Netherlands, with this master thesis work. It was a great journey and learning experience at UT with excellent faculty members. I would like to sincerely thank all my Professors at UT for sharing their knowledge and sowing the seed of research in me during my study at UT. I would like to extend my sincere gratitude to all the people involved directly or indirectly in this project work. This master thesis is carried out at NXP Semiconductors, located at High Tech Campus, Eindhoven, The Netherlands. It has been a very pleasant experience working at NXP as a master thesis intern with highly experienced and knowledgeable colleagues.

I would like to thank especially:
Dr. Frank Kargl, my direct supervisor at UT, for giving me valuable guidance during the thesis by means of timely instructions, helpful resources and feedback on my work.
Hans de Jong, my supervisor at NXP, for giving me an opportunity to do my master thesis at NXP and for your supervision, help and support right from the beginning of this thesis work. Thank you very much for providing feedback about my work but also giving me opportunity and freedom to think on my own and implement my ideas.
Cas Groot (NXP) for coordinating between UT and NXP, suggesting this project for my master thesis and for the support throughout my thesis work at NXP.
Pim Vullers (NXP) for the time, patience and enthusiasm with which you discussed and shared your knowledge about IRMA. I thank you for guiding me at all points during my research work at NXP, reviewing my work from time to time and giving new ideas about possible applications and taking this work to next step.
Stefan Kuipers (NXP) for the important discussions about the eID systems and sharing your experience.
Prof. Bart Jacobs (RU Nijmegen) for his cooperation and providing me an opportunity to be a part of regular IRMA meetings.
Gergely Alpár (RU Nijmegen) for your fruitful discussions regarding IRMA that gave me ideas for improvising my work.
Antonio De La Piedra (RU Nijmegen) for reviewing and providing valuable feedback that helped me in improving my report.
All my friends and colleagues at UT and NXP, Eindhoven for their support.
EIT ICT Labs for starting the double degree masters program and providing me an opportunity to study with full scholarship at two top technical universities in Europe (University of Trento, Italy and University of Twente, the Netherlands).
Last but not the least I would like to thank my parents for their unconditional support during my studies and stay in Europe.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In today's world, the use of the Internet has become more widespread than imagined before as people use computers and the Internet not just for gathering information or fun but also for carrying out important daily activities such as banking, shopping, social interaction and many others. The systems and the infrastructure for such activities in the physical world are also making a transition to the digital world in different forms. Just like the paper mail was overtaken by the email, paper tickets for public transport are gradually being replaced by electronic card and the paper identity documents such as passports have been equipped with digital chips to hold the digital copies of the identity data of the individuals. The electronic identification systems aim to provide universal, unique and reliable identification and authentication mechanisms to the citizens. Many governments have already introduced or are about to introduce electronic ID cards to its citizens. The increasing number of eID infrastructures and initiatives taken to scale the eID support to the eGovernment and eCommerce services have necessitated the security and privacy to be of highest order in all the use-case scenarios.

Self-assigned passwords or sending a password by snail mail on the creation of a personal account is not sufficiently strong identification or authentication means. A comprehensible verification of identity has so far taken place offline for example, by post or physical verification of the identity card. Even online processes sometimes require an offline interaction for identity verification. For example, opening a bank account at a local bank or via Internet requires the individual to furnish a proof of identity for which an ID card can be presented at the bank. Anybody ordering from a web shop must disclose their identity. The same applies when booking a trip, transferring money online or accessing an eGovernment service. The online social networks and forums require that their customers disclose more and more personal data and hence their reveal their complete identity on the Internet. All of this data is frequently not really necessary for a transaction. In addition

to that, prevalent data handling and security mechanisms and data protection legislation can be questionable when it comes to sensitive data. Furthermore, most of the existing digital systems use unique identifiers in order to identify the individuals using the system; this may be helpful for accountability purposes but it compromises the individual's privacy to a great extent. It becomes very easy to trace the user and gain information about the transactions carried out by her. For some transactions, identification is not even necessary; for instance, when one has to prove she is above 18 years of age to buy liquor at a store or when one has to prove she has a valid ticket before boarding a train, it does not matter who she is but what matters is if she fulfils a particular condition. So there is no need for the system to know the identity of that person. A more privacy friendly approach would have been to check for only such 'attributes' or characteristics belonging to a person which is required to complete the transaction. In the above examples, the attributes would be 'age above 18 years' and 'has a valid ticket' respectively. Do you absolutely have to provide more information than required? There is a need to minimize the unnecessary collection, sharing and disclosure of identity information. This in turn can reduce the instances online frauds such as identity thefts.

As mentioned earlier, many national governments are undergoing a change over from paper-based IDs to electronic ID cards in order to achieve a heightened security, higher government transparency and increased flexibility for Internet-based transactions. A position paper issued by ENISA on "Privacy Features of European eID Card Specifications" [3] underlines the need for "privacy-respecting use of unique identifiers" in the emerging European eID cards and countries like Germany have taken this into consideration in their national eID card's design and deployment. The German eID card *neuer Personalausweis (nPA)* which is by-far the most advanced and privacy preserving eID solution till date has taken extra care in the direction of improved user control, data protection and privacy. But how far does nPA satisfy its original security and privacy objectives? Can it be made better? These questions are addressed in this thesis where the main focus is to analyze this nPA system, its eID functionality and draw out the merits and demerits of nPA from security, privacy and usability perspectives. If the *'attributes'* of an individual can prove that she satisfies the required criteria for an activity then such a system would provide more privacy. Here, we introduce the Attribute-based credentials (ABC) and an ABC pilot project IRMA (I Reveal My Attributes) that is a smart card implementation of ABCs. It stores the attributes of the cardholder inside credentials and its main focus is to preserve the security and privacy of the personal data of the cardholder stored in the card. IRMA is a partial imple-

mentation of Identity Mixer[1] (Idemix) which is an anonymous credential system developed by IBM Zurich. Idemix allows the users to minimise the personal data they have to reveal in the transactions that require identification or authentication. It makes use of "*credential as a secure container for attributes*" concept and enables strong authentication and privacy at the same time. In this direction, an EU-funded research project ABC4Trust[2] has focused on the implementation of trustworthy and privacy-preserving identity management systems that support Attribute-based credentials. This project consists of a library over the Idemix implementation that separates the different entities of a ABC system: user, verifier, issuer, revocation agent, inspector. The deliverables[3] from ABC4Trust gives us an overview of these entities and describe how an ABC system is deployed in practice [4]. The FutureID[4] project is an initiative for building comprehensive, flexible and privacy-aware identity management systems in Europe and providing a common layer of authentication across Europe by enabling an integrative framework between different eID infrastructures and emerging trust service providers. The FutureID also supports the use of attribute based credentials. The recent research in Attribute based credentials [5] [6] has demonstrated ABC's improved performance on smart cards and their potential to become suitable options to be integrated into privacy-preserving eID solutions. In this thesis, we research the feasibility of merging IRMA and nPA in order to overcome the existing drawbacks of nPA with a view to make it more flexible and privacy-preserving.

1.0.1 Identification and authentication

Identification is about making a claim that a person is somebody. When a person introduces herself or answers the phone with her name, she has just identified herself. In the digital world, it is analogous to entering a username on a website. But it is not analogous to entering a password because a person verifies her username-claim by entering the password that is known only to her. This verification is termed as authentication where a person proves that she is indeed who she claims to be. Another form of authenticating one's self is by presenting an ID card (smart card) that contains the biometric properties of the person for instance, photo, fingerprints etc., By presenting this card, the cardholder can prove her authenticity. Based on the result of this authentication, the system will allow or not allow access to the cardholder. The authorization levels for that user are also checked/decided by the authenticating system to regulate the cardholder's access to the resource.

[1] http://idemix.wordpress.com/
[2] https://abc4trust.eu/
[3] https://abc4trust.eu/index.php/pub/deliverables
[4] http://www.futureid.eu/index.php/about

Different kinds of authentication

In [7], *Entity authentication* is defined as the process whereby one party is assured (through acquisition of corroborative evidence) of the identity of a second party involved in a protocol, and that the second has actually participated (i.e. the party is active at or immediately prior to the time the evidence is acquired) in the communication. The method of entity authentication in the case of user authentication varies between the widely used password verification, PKI based certificate verification, challenge-response authentication and biometric recognition. These categories of authentication enable direct authentication where a unique ID is a mandatory requirement, otherwise the user to be authenticated is not identifiable. Here it is obvious that the user is directly recognized and her transactions can be tracked as there is the unique ID for the reference and thus user's privacy is lost completely.

Message authentication is a means to make sure that the message is from the claimed originator and that the integrity of the message has not been tampered during transmission [7]. The conventional approaches for message authentication requires the originator to reveal its unique ID, since otherwise there is no way to link the message to the originator. As mentioned earlier, use of unique ID is unfavourable if privacy protection is a criterion. Moreover, in certain situations it is desirable that the messages sent by same sender are indistinguishable from messages sent by other users. This privacy requirement is termed as *anonymity*. In some situations, it is desirable that two or more messages sent by the same sender cannot be linked to each other leading to profiling/tracking of the user. This privacy feature is termed as *unlinkability*.

Pseudonym authentication: A pseudonym is an arbitrary identifier of an identifiable entity by which a certain action can be linked to this specific entity [8]. *Pseudonym-based authentication* is a variant of entity authentication where pseudonyms (non-real names or random numbers) can be used by the users rather than using their real names or unique identifiers for authentication. The advantage of this method is the privacy-protection of the user to an extent as her real identity is hidden from all the parties that are authenticating her. In case of PKI, a pseudonym is often the public key present in the PKI certificate.

Attribute Authentication: Attributes are usually the properties or characteristics of a person/entity. Certain attributes or a combination of some attributes can authenticate the person instead of directly authenticating the identity of the person. So in a communication, attribute authentication does not need the actual identity of the participating entity. For example, Person X is older than 18 years of

age so she can purchase liquor offline or enter an online gambling site; when she has to authenticate herself to the liquor store or the gambling site, only 'age' attribute is required not her full identity (including name, address, nationality etc.,). Attribute authentication facilitates such data-minimized authentication and achieves user anonymity. In conclusion, it provides higher flexibility of authentication and privacy for the users when compared to the entity authentication.

1.0.2 Main functions of an eID system

Internet being one of the current modes to carry out the governmental and business transactions makes a person's identity (or specific user-identity attributes) play a crucial role when involved in such transactions. Many of the services are already available on the Internet and many more will be available in near future. So, it is very important to know the counterpart in a communication i.e. the service that the user is communicating with is legitimate. It is not possible to physically call a meeting in order to authenticate the entities involved in an online transaction since this scenario takes place in cyberspace. The user becomes the *prover* who is supposed to prove his identity to the service provider who then becomes the *verifier*. Each participant involved in a communication must authenticate herself to the others and the prover must be able to control the revelation and flow of her personal identity data. Also, the physical documents cannot be used here to transfer trust in identity information from the identity document issuer to the verifier. Thus there is a need for a secure electronic identification system which satisfies all the online security requirements. Especially, if such an identification is being done using a national electronic ID document, an extreme reliability on the eID system is expected and henceforth such a system should have security and privacy enhanced functionality with a high level of trustworthiness. A national eID document is supposed to securely identify and authenticate citizens of a country both online and offline, it should provide certain functionality which are summarized as follows [9]:

- **Visual identification and verification**: An eID cardholder can be physically identified with the aid of the cardholder's picture on the eID card and the information printed on the card. If the biometric information is also stored on the eID card chip, it can be used for identifying an individual at the national borders and check-posts where the biometric data from the person and the eID card can be read by the authorized terminals and matched.
- **Secure digital identification and authentication**: While accessing any online service, a user has to identify herself (with name/username) and authenticate to the service (proving that she is really who she claims to be).

This can be done either with a username-password combination or with a smartcard using cryptography based on the required security level. The chip present in the eID card contains a number of data groups comprising of the cardholder's identity, address, electronic signature and optionally biometric signature files. These files are sufficient to digitally identify the cardholder in the context of an online transaction.

- **Digital Signature**: With this functionality, the citizens can sign the documents digitally with a Qualified Electronic Signature[5]. Once enabled, the digital signature application contains a registered certificate and a private/public key pair and if the user wishes to e-sign a document, she has to first authenticate to the card using her secret PIN. The card then signs the document (or its hash) and this e-sign can be used to prove the integrity of the document. This feature of eID card can be used online or offline.

- Data privacy protection and increased control of the eID cardholder over her personal data that can be disclosed about her and to whom. This can range from simple PIN protection to sophisticated certificate-based access control mechanisms or domain-specific identifiers.

- EMV standards (Europay, Mastercard, Visa) define a payment method for the smartcards. Some governments are associating payment services along with the national eID card [10].

1.0.3 Thesis objectives and structure

The main objective of this thesis is to research if Attribute-based credentials can be integrated into an eID system like the German eID (neuer Personalausweis) to enhance privacy and flexibility of the digital authentication functions. In order to fulfil this objective, we carry out the following steps through the course of this thesis work:

1. Analyse the main features and functionality of eID systems.

2. State the important technical requirements for the national eID system.

3. Provide a detailed description of German eID neuer Personalausweis (nPA), its functionality and the cryptographic protocols used in nPA's eID authentication function.

[5]EU Directive for electronic signatures: http://europa.eu/legislation_summaries/information_society/other_policies/l24118_en.htm

4. Identify the drawbacks of nPA from a technical perspective especially in terms of privacy and flexibility.

5. Introduce the attribute-based credential design which is used by the IRMA (I Reveal My Attributes) technology while detailing its privacy enhancing features.

6. With an aim to make the eID system more flexible, secure and privacy-preserving, conduct a thorough analysis on the feasibility of incorporating IRMA in eID authentication and merging it with nPA; design the protocols and system architecture to suit the proposed IRMA-based eID system.

7. Investigate the advantages and drawbacks of IRMA based eID authentication approach.

8. Analyse the performance consideration of Idemix implementation of IRMA and compare it to nPA eID authentication function's overall performance.

9. Enumerate few use-case scenarios where the proposed system will serve as the best-fit for the user authentication purposes.

The thesis is structured as follows: In the chapter 1, we provide an introduction to the advent of electronic identification systems, issues arising from such Internet-based identification systems and list the general requirements for a secure and privacy-preserving electronic identification and authentication system. In the chapter 2, we describe the German eID system (neuer Personalausweiss), its eID functionalty and the involved security protocols in detail. In the same section, we carefully analyse, identify and discuss about the main limitations of the German eID system. Then we move on to the chapter 3 where we introduce the Attribute Based Credentials (ABCs) and the IRMA project which is the partial implementation of the Idemix credential system. In this chapter, we describe the main features of IRMA such as selective disclosure of attributes, data minimization, pseudonym generation and revocation of the credentials. In the chapter 4, we propose our new IRMA-based authentication scheme that integrates IRMA into the German eID system for the purpose of eID authentication and analyse the security, advantages of this scheme over the German eID system and also its disadvantages. Some relevant use cases that could make use of the proposed IRMA-based authentication scheme are also discussed in this chapter. Finally, in the chapter 5, we conclude our work and mention some recommendations.

1.1 Requirements for an enhanced eID system

In this section, we postulate some general requirements for an enhanced eID system in terms of security, privacy and usability. As we consider the German eID model is be the most advanced eID deployed in recent times in terms of security and privacy, we take inspiration from some of its security features [11] while drafting the requirements for a more enhanced eID system than the German eID system. Recently, the Dutch government has been involved with the formulation such an eID system[6] and we also refer to some of the design requirements stated in their documentation [12] that we find relevant to the context of this thesis.

1.1.1 Security and privacy requirements for a national eID system

The security and trustworthiness a national eID system have to be increased in the interest of all the stakeholders involved in the system such as users, service providers, public and private entities involved in the set-up, operations and maintenance of an eID system. Several aspects of security that needs to be addressed in an eID system are specified as follows:

1. Increase *security and trustworthiness* by
   - Enabling greater user control over authorizations and data.
   - Drafting measures in which the service providers can be held into account in the case of data misuse or any other fraudulence.
   - Restricting data collection extent to the minimum limit that is absolutely necessary.
2. *Authenticity*: Enable authentication of the participants involved in an eID-related communication.
   Mutual authentication: Along with the user identification, mutual authentication should be implemented for all important security centred online transactions where both user and the service provider have to prove to each other who they claim to be. This is to ensure that users also are guaranteed of communicating with an authentic party on the Internet. Reason: If only the user has to authenticate and there is no service provider authentication, the user can not be sure if she is communicating with a legitimate service provider or the adversary. This unawareness of the user might lead to identity theft or abuse of information if information falls into the wrong hands.

[6] http://www.eid-stelsel.nl/over-eid-stelsel/

3. *Increase confidentiality and integrity*: Advanced cryptographic protocols with strong keys must be used to encrypt the communication between the user and the service provider and to verify the integrity of the messages exchanged between them during an authentication session. All the channels between the prover and verifier must be secured with strong cryptographic primitives and the use of weak cipher suites must be forbidden; for instance, if the TLS channels are being used in the context of an eID online authentication, a set of allowed cipher suites must be restricted and hence effectively preventing non-encrypting cipher suites. This requirement will ensure that the network channels are not intercepted by any adversary. If this requirement is not fulfilled, an adversary can eavesdrop on the ongoing communication between the user and the verifier (e.g. service provider) thereby, affecting the confidentiality and subsequently harming the privacy of the exchanged information during that communication.

4. *Provide privacy protection*: The leakage of any personal or privacy-infringing information about the sender of the messages must be prevented; Linkability of information collected during card issuance and data verification instances must also be strongly prevented as it might lead to undesired disclosure of transactions made by the user eventually leading to her complete profiling. The identity management server (for example, eID-Server in nPA) must be deployed at the service provider domain instead of having third parties running these servers; alternatively, the eID-Server can be made stateless and not keep any logs of the interactions it is involved with. However, this should be constantly monitored by the authorities. If the third parties are running these eID servers, then the right sort of legal framework must be setup around it in order to enforce security and privacy policies. This is to prevent the privacy risks incurring from the eID-Server seeing all attributes it verifies, for instance, the eID-Server will then know which attributes were meant for which relying party and it will also be able to track the traffic patterns between the users and the relying parties.

5. As mentioned in the first requirement, *maximum user control* over her personal information being transmitted is desired. It means that the users should have full control to selectively reveal their personal data (or a part of their data) stored on the the card or even certain properties of such attributes (like, age ≥ 18).

6. *Qualified Electronic Signatures (QES)* scheme must also be supported by eID cards by adding higher levels of security to add a legal certainty to online correspondences. The digital signature or the QES creates a legal

relationship between the user and the service provider in concluding contracts or taking decisions.

7. *Non-repudiation*: This is required in the case of some tax/financial fraud or disputes. When the inspection authorities want to examine if this transaction involved a particular eID card, it must be possible. If non-repudiation is desired, an eID cardholder must not be able to deny that she had sent a message or authenticated an attribute if she had actually done it. This requirement is implicitly taken care of, if the transaction involves the digital signatures of the parties involved.

8. *Blacklisting or revocation* of authentication tokens (e.g. smartcards) and rogue service providers must be facilitated in a secure manner without compromising any of the privacy-preserving objectives. Revocation becomes necessary even in the cases of lost, expired or stolen eID cards.

9. *Minimization of dependencies* between the components must be encouraged so as to eliminate bottleneck situations. For instance in the German eID system, the entire security of the eID system is bound to the secure element on the chip and chip authentication keys shared among a large group of eID cards. If one card is compromised then all the cards in the group must be blacklisted. Such risks must be minimized when it come to an eID system at a national level.

1.1.2 Usability requirements

An eID system can be successful only if it is used and adopted widely by the users. So usability criteria plays a major role in assessing the performance of an eID system. Henceforth, we list out some usability requirements [12] [13] for the eID system:

1. The eID system and its functions must be designed in a user-friendly and accessible manner such that it can be used by all its users even the ones with less digital experience.

2. The key values and advantages of such a system must be clearly elicited to its users and many digital services that support eID functions should be easily available to them.

3. Minimal or at least reasonable processing time and computational overhead is expected from such an eID system for authentication and signature functions to enhance the user experience.

4. Participation of the private parties (i.e. private digital service providers) should be encouraged and facilitated by an eID system along with the public parties (i.e. eGovernment service providers). The users will use the eID system more if more services support authentication with the eID card.

5. Allowing the users to control the transmission and authorization of their data is crucial even from the usability perspective as it provides a sense of control to the users thus, making them confident to use such a system.

6. The users must be given an option to save their eID-related transactions for accountability purposes and check which data is present on their card.

7. In the case of lost eID tokens (e.g. eID card), the transition from old to new eID tokens must take place in a reasonably short time and if the pseudonym features are supported by the eID card then there should be a possibility to securely carry forward all the previous transactions done by the user under that particular pseudonym.

Chapter 2

The German eID - nPA

2.1 Introduction to the nPA functionality

Germany introduced new personal identification card (neuer Personalausweis - nPA) for its citizens on November 1, 2010. This identity document is an electronic, multifunctional card in the credit-card format, valid as a travel document and as proof of identity both in the physical and the electronic world. The German electronic ID (eID[1]) card not only provides the conventional passport-enabled identification (ePass function) for exclusive governmental use (e.g. Border control with authorized inspection terminals) but it is also equipped with two new electronic functions:

- Mutual electronic proof of identity (eID function) for citizens who require tro authenticate to certain eBusiness and eGovernment applications. For example, if an online shopping and delivery service requires to verify the place of residence of the user then bothe the user and service can mutually check each other's identity in a secure manner.

- Qualified Electronic Signature (eSign function) for the citizens to place their electronic signatures on the documents that require legal certainty. For example, an eID cardholder can digitally sign and file a tax declaration form online.

The combination of a sovereign identity document with the eID functionality aimed to provide the users with a secure identity in the electronic world with better protection against many types of cybercrime, such as phishing and identity theft [1]. The personal data being transmitted between the user and the service provider is self-determined by the user thereby enabling better user-control over her personal

[1]The terms 'nPA' and 'German eID' are used interchangeably throughout this thesis.

data and its protection. Data minimization functions are also included in the eID function which are described later in this section. Biometric data of the user stored on the eID card (for e.g. photo, fingerprints, eye color, height and personal signature) are never transmitted to the service providers or via the Internet. Only the sovereign authorities who possess the authorization and the technical means can query such sensitive information. Figure 2.1 is the visual representation of the German eID card[2] and table 2.1 provides an overview of its main functions.

Figure 2.1: German eID card (nPA)

The important design goals of nPA were: privacy, data security and citizens' ability to know the extent of access that the external services have to their own personal data. Thus it protects against the misuse of data and identity theft as well as supports the principles of data avoidance and minimization. The citizens can use this card to securely access both online business services and also offline services (non-Internet) such as for hiring cars or at ticket vending machines. Biometric data is strongly protected; it can be only read when the eID card is physically present and can only be accessed by the official authorities such as police, tax authorities and so on. The unique feature in the German eID card is that the data on the contactless chip is fully under the control of the card owner and the service providers must authorize to the eID card before any data can be read from it. The service provider can prove its authorization via a card verifiable certificate (CVC) and the eID card owner has to further grant permission to such a service provider by entering a secret 6-digit PIN. The eID card consists of data minimization functions such as age verification, residence verification and restricted identification in addition to the data as mentioned in the Table 2.1. The objective of these functions is to transmit from the user to the service provider only the minimum data

[2]More visuals on the German eID card format and data present on it can be found at http://en.wikipedia.org/wiki/German_identity_card

Electronic functions of nPA			
eID Functions	**Purpose**	**Data**	**Special Functions**
ePass (mandatory)	Readout by authorized offline inspection systems	- Face image - 2 fingerprint images (optional) - MRZ data	None
eID (activation optional)	Online applications read data or access functions as authorized Offline inspection systems read all data, update address and community ID	- Family name, given name - Artistic name, doctoral degree - Address and community ID - Date and place of birth - Date of expiry	- Age verification - Residence/Community ID verification - Restricted identification (pseudonym) - Revocation feature
eSign (certificate optional)	Certification authority installs signature certificate online Citizen makes electronic signatures with eSign PIN	- Signature key and X.509 certificate	- Create electronic signatures

Table 2.1: Overview of electronic functions of nPA

that is needed for that specific transaction. The working of these functions work is briefly described below:

1 **Age verification** function by which the service provider (SP) will get the information about a user's age, for instance, if a certain age is reached, say 16 or 18 years, based on a reference/test date sent to the chip of the eID card by the service provider. This function requires the proof of the *age verification right* by the SP terminal to the card. The test date is actually the date *required* by the SP terminal and the chip on the eID card will compare this test date to the stored date of birth. The age verification is successful if the stored date of birth is not after the required date of birth. The output of

the age verification function is a simple 'yes' or 'no' answer that is returned to the service provider instead of actual birth date of the eID cardholder. The test date is sent as a part of the Terminal Authentication (explained in Section 2.4.2) and verified by the chip in order to prevent targeted narrowing down of the cardholder's age by sending repeated queries to the card with different test dates.
During TA, the test date is sent as a part of the auxiliary data which is signed by the SP terminal and verified by the chip. So this auxiliary data is authenticated at the end of TA. As it is mandatory for TA to be executed only once within a secure session, the test date can be sent only once to the card chip within a session. If the SP terminal wishes to send the test dates many times, it has to establish a new secure session each time with the card. This necessitates the user to enter her PIN everytime in order to initiate PACE and then TA. So, in conclusion, it is not feasible to find the actual age of the cardholder by sending different test dates repeatedly to the card.

2 **Residence verification** function by which the service providers can verify if the eID cardholder lives in a particular region or a city. This function does not transmit the full address of the cardholder but just compares the regional identifier (or the official municipality code number) of the cardholder's address against a reference regional identifier sent to the card by the service provider and a simple 'yes' or 'no' answer is returned. The official municipality code number contains the information consisting of the land (i.e. Bundesland), the administrative region, city or district and the municipality. To provide regionally or localized services to the citizens, the service providers can send query concerning the place of residence and this residence query also enables a query corresponding to other levels such as land, administrative region or district. This is helpful for the service providers who would like to offer services for the inhabitants of a particular administrative region only. As in the age verification procedure, the place queried is transmitted as part of the Terminal Authentication data so that it is not possible for a service provider to narrow down the place of residence by sending repeated queries to the card.

3 **Restricted Identification or pseudonyms**: With this function, an eID cardholder can identify herself without revealing any personal information i.e a false name or a pseudonym can be used to access a service. nPA supports sector-specific pseudonyms where a sector could be any public or private sector (e.g. Healthcare sector). A pseudonym in nPA is a sector-specific identifier that is calculated for every *user-service provider pair* by combining the secret key belonging to the eID card and the unique identifier of the

service provider that is sent as a part of its authorization certificate. Thus, a service provider will be able to recognize the user as the owner of a specific eID card when a service is accessed using the pseudonym. However, the user has to provide her consent for the access of her pseudonym by the service provider by entering her eID PIN. This function is useful in the instances such as, when a user wishes to enter an online forum or participate in an online survey without having to give away any of her personally identifying data during the registration. The pseudonym generation procedure in nPA is discussed in more detail in the Section 2.5.

2.2 The eID function in nPA

The **electronic identification or the eID function** is one of the three major functions of the nPA as mentioned earlier in the Section 2.1 (Also, mentioned in Table 2.1). With the activation of this eID function, nPA can be used for the mutual identification and authentication of the citizens and the Service Providers (SP); this ensures both parties to know who the counterpart claims to be. Electronic authentication is meant to establish a trusted and a secure channel between the eID card chip and the provider. The card owner proves her authenticity by the possession of card and secret PIN whereas SP proves its authenticity with the aid of an authorization certificate. Technically, only the service providers who have the approval of Federal Office of Administration (Vergabestelle für Berechtigungszertifikate)[3] can read the data from any eID card. The service provider is responsible for protecting the data that is read from the eID from misuse or falling into hands of third parties.

The German eID card infrastructure consists of several components both on the user and service provider sides. Typically, a user needs the following components in order to use the eID function during online transactions: (1) an eID card with an activated eID function (2) a local card reader (3) The eID-Client application (Ausweissapp software).
A service provider (SP) needs the following components in place if he wishes to integrate the eID function into the existing services: (1) an offline or an online service portal with an authorization certificate that grants the service provider a controlled access to the data on the eID card (2) an eID-Server (attached SP eID server or a third party eID server) that handles the authentication by communicating with the eID card using the cryptographic protocols.

[3]Vergabestelle für Berechtigungszertifikate (VfB) : A new government institution, the Issuing Unit for Terminal Certificates which is part of the Federal Office of Administration (Bundesverwaltungsamt – BVA), issues Card Verifiable (CV) certificates to the service providers.

The online eID authentication process is described as follows:

1. The user visits the an online website *eService* and requests a service from the service provider (SP). The SP requires the user to authenticate with her eID card.

2. The eService sends a webpage with an embedded link[4] and all the necessary parameters to the user's browser in order to establish a secure connection between the eID-Client and the eID-Server (Identity Provider).

3. The user clicks on the link and the user's browser (User-Agent) performs a corresponding HTTP GET command to start the eID-Client application (Ausweisapp).

4. The eID-Client application establishes a TLS channel to the eID-Server and displays the user's data requested by the SP to the user on her browser. The citizen decides which data the SP is allowed to obtain. By entering the PIN, the citizen gives her consent for the data access and transmission. Another secure messaging channel is setup on the top of this TLS channel by the session keys generated during Extended Access Control (EAC) protocols; Within this secure and trusted channel, the actual authentication takes place and the required user data/attributes are sent to the eID-Server.

5. The eID-Server verifies the user attributes and performs user authentication. It conveys the result of this authentication to the service provider and optionally sends the user attributes if requested by the service provider; but the attributes are sent only if the service provider has the rights to read those attributes and the transmission is permitted by the user. On a successful authentication, the service provider grants access for the service to the user.

The communication between the technical components of nPA during the eID authentication is shown in the figure 2.2.

2.3 Realization of electronic authentication function in nPA

Many eID cards in the existing eID systems are equipped with X.509 certificates for TLS client authentication [14]. But when the X.509 certificate is transmit-

[4]It is an embedded link pointing to the following URL `http://localhost:24727/eID-Client?=tcTokenURL=..`, which instructs the eID-Client to pull the required address information and the corresponding X.509-based TLS server certificate, which is to be checked against the CVC (Card-Verifiable- Certificate) of the eID-Server.

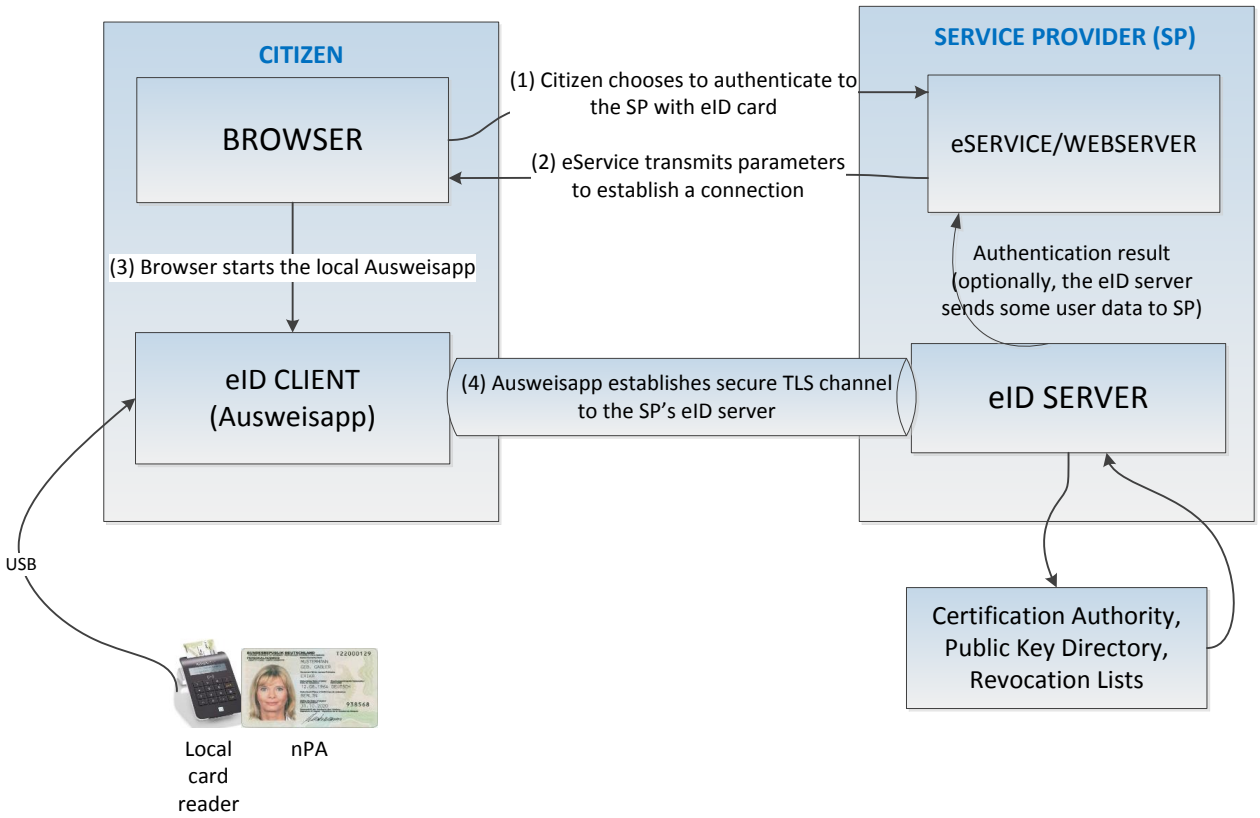

Figure 2.2: Online authentication procedure with nPA

ted over the unprotected channel during the TLS handshake, the user's privacy is lost as some sensitive identity information contained in these certificates like the cardholder's name is exposed. Due to this reason, nPA does not use X.509 certificates for the TLS authentication, instead, nPA uses Extended Access Control mechanisms which makes nPA more secure and privacy-friendly. In this section, we describe the infrastructure and cryptographic primitives that are used as a part of nPA online authentication.

The EAC protocols are explained in Section 2.4. To realize EAC and guarantee the authenticity of eID cards and the service providers, two Public Key Infrastructures (PKI) are used in nPA and they are briefly discussed in the following section.

2.3.1 PKI Infrastructure used in nPA

The German eID document makes use of two Public Key Infrastructures (PKI) for realizing its electronic functionality in a secure manner:
(1) Country Signing Certificate Authority (CSCA) for verifying the authenticity of the eID documents (during Passive Authentication).
(2) Country Verifying Certificate Authority (CVCA) for protecting the biometric data stored on the eID document and verifying the authorization and the access rights of the service providers (during Terminal Authentication).

Country Signing Certificate Authority (CSCA)

The CSCA is operated by the Federal Office for Information Security (BSI). The CSCA generates the German root certificates (CSCA certificates) on a regular basis, which in turn serve as the source for the private keys of the document signing certificates of the ID card manufacturer. These private keys are used by the authorized ID card manufacturer to sign the data files on the eID document. The document signing certificate is also electronically stored on the identity card. Using the CSCA root certificate, it is possible to verify whether an electronic identity card was indeed created on behalf of the issuing nation and whether the data on the card have been changed in any way since production. This is realized using Passive Authentication [1].

Country Verifying Certificate Authority (CVCA)

The CVCA is also operated by the BSI. It is the authority that generates the German root certificates whose private keys are used to sign the document verifier certificate of the certified document verifiers (DVs). The DVs are responsible for issuing the certificates authorizing the service providers for reading the electronic identity documents and also define the individual read rights, i.e. what information can be read from the identity documents. This authorization is verified by the eID card's RF chip during Terminal Authentication.

The Figure 2.3 illustrates the spectrum of variants of national authorization certificates for the new identity card. In addition to the applications for sovereign purposes i.e., ePass function and for electronic identification, the CVCA also supports the eSign function of the eID card for creating a Qualified Electronic Signature [1].

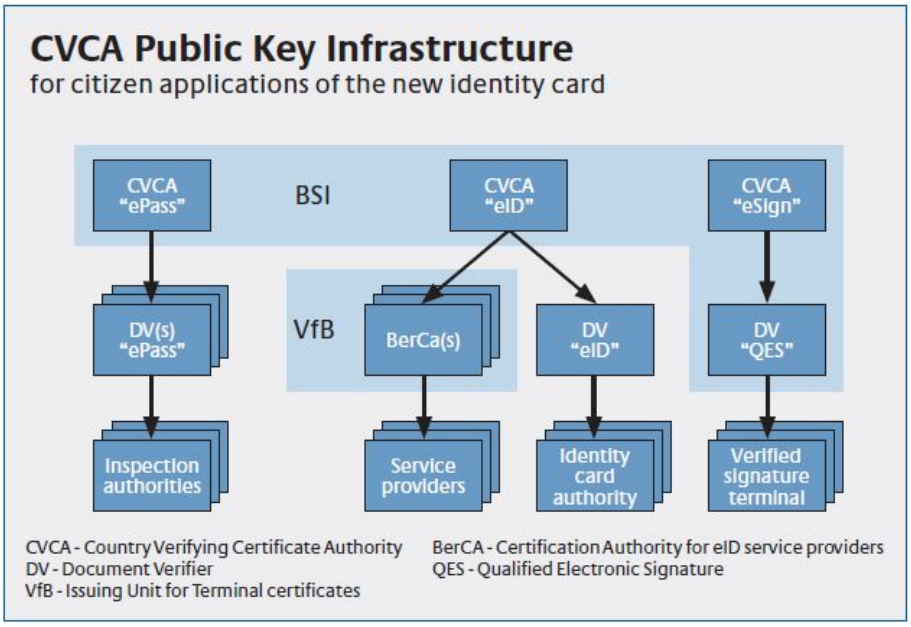

Figure 2.3: CVCA PKI for citizen applications of the eID card [1]

2.3.2 Passwords used in nPA

In nPA, Password Authenticated Connection Establishment (PACE) protocol is used as a secure and a practical mechanism to restrict access to the applications based on a particular knowledge, i.e. based on passwords that are either printed on the document or only known to the legitimate holder of the document. The detailed description of PACE can be found in the Section 2.4.1. The passwords used during the PACE protocol for reading the data from the card are listed in this section with respect to ePass, eID and eSign functions of nPA [2].

- The eID supports the ePass function where physical identification is carried out by the authorized official inspection systems. For this ePass function, **Card Access Number (CAN) or Machine Readable Zone (MRZ)** printed on the card is used as the password.

- For online scenarios of the eID function of nPA, **eID PIN** is used as password. PIN is the Personal Identification Number (PIN) which is a short secret password that is known only to the legitimate holder of the eID card. It is used to access eID application and the eID cardholder may allow an authentication terminal to access data stored on the eID application by entering this PIN unless the terminal has the effective authorization to access the eID-application with the CAN. The PIN is a blocking password, i.e. the

PIN is associated with a retry counter that is decreased for every failed authentication. To access the unblocking mechanism of the PIN, there is a password, PIN Unblock Key (PUK) which is a long secret password that is known only to the legitimate holder of the eID card. If the PIN is blocked and the user sucessfully enters the PUK, the PIN gets unblocked. PUK is a non-blocking password; however, it may be associated with a usage counter that is decreased for every successful authentication.

- For the card owner to exercise the eSign function, the certification authority installs the signature certificate online and eID PIN is the password for authenticating the card owner to the CA. Once the signature certificate is installed, the eID card owner can sign the documents electronically by using **eSign PIN**.

2.3.3 Keys used in EAC protocols and cryptographic primitives used in nPA

The keys and operations for the symmetric key encryption and authentication, their notations are briefly described below in a algorithm-independent manner. Symmetric keys are derived from a shared secret K and an (optional) nonce r or from a password π using a Key Derivation Function (KDF)[5]. The KDF is used to derive encryption keys and MAC (Message Authentication Code[6]) keys. The technical guideline BSI TR-03110 [2] summarizes the keys to be used in nPA as follows:

- Deriving a key for message encryption is denoted by $K_{Enc} = KDF_{Enc}(K,[r])$.
- Deriving a key for message authentication is denoted by $K_{MAC}=KDF_{MAC}(K,[r])$.
- Deriving a key from a password π (suitable password is chosen based on the terminal type; for more information refer the previous subsection on the Passwords used in nPA) is denoted by $K_{\pi}=KDF(\pi)$.

The operations for encrypting and decrypting a message are denoted as follows:

- Encrypting a plaintext m with key K_{Enc} is denoted by $c = E(K_{Enc},m)$.

[5]The definition for a KDF given in the Wikipedia is: A key derivation function (or KDF) derives one or more secret keys from a secret value such as a master key or other known information such as a password or passphrase using a pseudo-random function.

[6]A message authentication code (often MAC) is a short piece of information used to authenticate a message and to provide integrity and authenticity assurances on the message. The keyed hash function or cipher-based function that takes MAC key and message as input and outputs the MAC is known as the MAC function.

- Decrypting a ciphertext c with key K_{Enc} is denoted by m = D(K_{Enc},c).

The operation for computing an authentication code t on message m with key K_{MAC} is denoted as t = MAC(K_{MAC},m).

The KDF, Encryption, MAC and Key Agreement functions used in nPA are implemented using the following cryptographic primitives [2] [15]:

- The KDF used is the hashing function **SHA-256**[7].
- Encryption function used is **AES-128 CBC**[8] (AES-128 in Cipher Block Chaining mode).
- MAC function used is **AES-128 CMAC**[9] (AES-128 with Cipher-based Message Authentication Code.)
- Key establishment in nPA is done using **ECDH**[10] (Elliptic Curve Diffie-Hellman algorithm).
- **ECDSA** (Elliptic Curve Digital Signature Algorithm) is used for authorization certificates and signatures.

An overview of optimal keylengths for the keys used in nPA's cryptographic algorithms for the years 2013-2015 in accordance with BSI recommendations is as follows:

- For asymmetric cryptographic operations, the keylength specified is 1976 bits (For long-term security level, 2048 bits is recommended for RSA operations).
- For symmetric key cryptographic operations, the keylength specified is 128 bits.
- For discrete logarithmic functions, the keylength recommended is 256 bits and group size is 2048 bits.
- For elliptic curve cryptographic functions, recommended keylength is 224 bits.

The above information was retrieved from the web source BlueKrypt Cryptographic Keylength Recommendation[11] under the BSI recommendations section

[7] Secure Hash Algorithm: http://tools.ietf.org/html/rfc6234
[8] AES-CBC algorithm: http://tools.ietf.org/html/rfc3602
[9] AES-CMAC algorithm: http://tools.ietf.org/html/rfc4493
[10] Information on ECDH and ECDSA can be found at http://www.ietf.org/rfc/rfc4492.txt
[11] BlueKrypt Cryptographic Keylength Recommendation: http://www.keylength.com

and further comparison of keylength recommendations by other standards with the BSI standards can also be found at the same website.

Table 2.2 enlists all the sets of keys used during PACE and Extended Access Control (EAC) protocols that take place in the nPA eID authentication process. In this table, we have redefined the notations of the keys mentioned in the BSI TR-03110 [2] and we use these keys' notations in the Section 2.4 while describing each of the EAC protocol specifications.

Protocol	**Keys on the card chip C**	**Keys for the Terminal T**	**Note**
PACE	PK_{Ce}, SK_{Ce}	PK_{Te}, SK_{Te}	All keys are ephemeral DH keys
Terminal Authentication	PK_{CVCA}	PK_{T}, SK_{T}	The chip verifies the certificate chain received from the terminal using the public key of the CVCA.
Chip Authentication	PK_{C}, SK_{C}	$PK_{Te'}$, $SK_{Te'}$	The key pair used by the terminal is an ephemeral key pair different from the ephemeral PACE key pair.
Restricted-Identification (Pseudonyms)	SK_{ID}	PK_{sector}	The chip should not provide the corresponding public key PK_{ID}, the terminal must not be provided the corresponding private key SK_{Sector}. The keys PK_{ID} and SK_{Sector} are externally used to generate revocation lists.

Table 2.2: Overview of key pairs used in EAC protocols in nPA [2]

2.4 EAC Protocols used in the eID Online Authentication

The Extended Access Control (EAC) mechanism comprises an array of protocols that are always executed in a specific order, depending on which electronic identity document is to be read [2]. The general authentication procedure in nPA can be briefly described in the following steps:

1. **Password Authenticated Connection Establishment (PACE)**: As the name suggests, PACE protocol is meant to authenticate the user to the card reader as the legitimate owner of the card by entering the PIN (Password); it is also meant to create a secure channel between the card and the reader in order to avoid the card being read from a distance without an explicitly granted access by the card owner. PACE starts the secure messaging between the card and the card reader only if the indication of the card reader/terminal type (inspection terminal, an authentication terminal or a signature terminal) and its requested access rights are successfully checked. Thus, PACE provides trust points for the next step Terminal authentication.

2. **Terminal Authentication (TA)**: During this protocol, the terminal sends the complete certificate chain starting with the CVCA certificate that is verifiable with CVCA public key and ending with the terminal certificate. Thus the chip can verify the authenticity of the certificates and extract the static public key of the terminal. The terminal generates the ephemeral public key to be used later on for the chip authentication. The chip sends a nonce, the terminal signs the nonce with its static private key which the chip can verify with the public key extracted from the terminal certificate. If the authentication is successful, then the eID card chip grants the read/write access rights to the data groups based on the terminal access rights. The terminal authentication also restricts those access rights to Secure Messaging to be established by the authenticated ephemeral public key.

3. **Passive Authentication (PA)**: PA is meant to verify the integrity of the data stored on the eID card. During PA, the terminal reads the unsecured security information from the chip's CardAccess file before PACE and the terminal reads the CardSecurity file after PACE and TA are executed. The security information that is common in both CardAccess and CardSecurity files are matched (one-to-one matching) and CSCA certificate and the signature over the security inofrmation in the CardSecurity file are verfied by the terminal. If the verification is successful, then the terminal is convinced that the card data has maintained its integrity.

4. **Chip Authentication (CA)**: In this phase, the authenticity of the chip is verified i.e. its is checked if the chip is forged or original. The terminal's ephemeral public key computed by the chip during CA is compared with the ephemeral public key generated by the terminal during Terminal Authentication. If it matches, then both chip and the terminal move ahead to agreeing on a shared secret key. Then the chip derives the session keys based on the shared secret key and generates an authentication token which is verified by the terminal in the next step. The chip is authenticated if this verification is successful. Then the chip restarts the secure messaging with the newly generated CA session keys; the security context established by this securing messaging will be used for all the further communication between the card and the authenticated terminal.

Figure 2.4 illustrates the simplified version of EAC protocols in the order as they are executed in nPA and the protocols are explained in detail in the further sections.

2.4.1 Password Authenticated Connection Establishment (PACE)

Any communication with the chip of the ID card can only be performed if the cardholder enters her PIN to the chip. This guarantees a so-called two-factor authentication based on ownership (the ID card) and knowledge (the PIN). As the chip is contactless, the PIN cannot be sent 'over the air' without additional protection [11]. The Password Authenticated Connection Establishment (PACE) protocol is designed for such an access control, mainly to protect the RF chip on the card to be read at a distance without an explicitly granted access. The national identity documents like the passports used Basic Access Control but in the case of nPA, PACE is used as an alternative to the Basic Access Control (BAC).

BAC uses the date of birth, the expiry date and the serial number retrieved from the MRZ on the passport as the password to verify physical access to the passport and to generate session keys for the protection (encryption and authentication) of subsequent communications. The BAC protocol is based on a mutual challenge-response sequence that relies on symmetric cryptography. BAC was introduced in order to prevent the skimming and eavesdropping on the data on the chip but it was designed for less sensitive data and with ease of implementation in mind. Thus, the security provided by BAC is adequate but not very good. We refer to Bender et al. [16] where some specific reasons for replacing BAC with PACE in THE German eID system are mentioned and two of the main reasons are stated below:

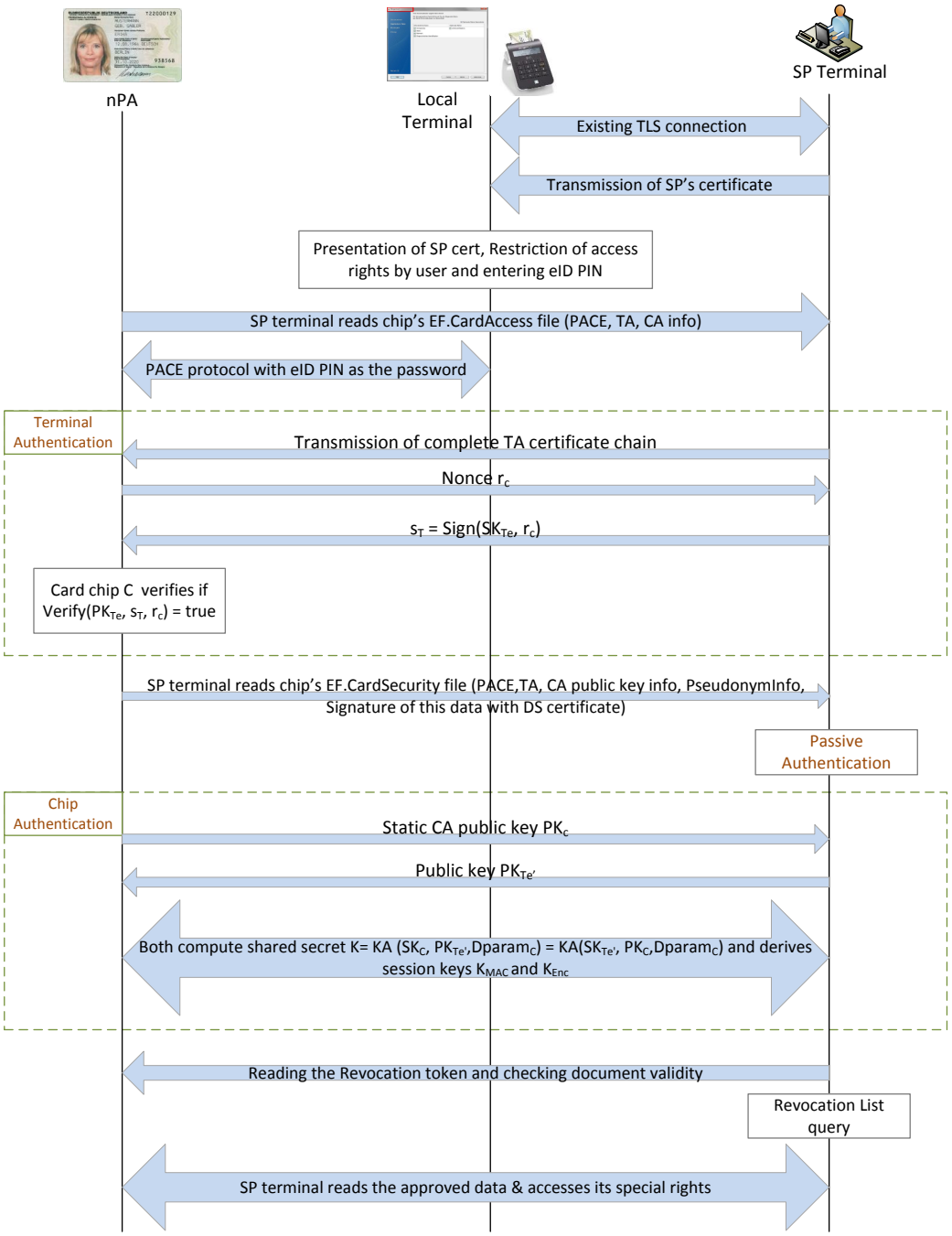

Figure 2.4: Simplified flow diagram of EAC protocols in nPA

- Due to the use of symmetric cryptography in BAC, the strength (entropy) of the keys used to encrypt and authenticate the contactless communication is held back by the limited strength of the MRZ-derived password. Depend-

ing on the numbering convention used for the serial numbers, the authors estimate the practical entropy of the MRZ is in between 50 bits (using alphanumeric, random serial numbers) and 40 bits (using sequential serial numbers).

- If the execution time of BAC is increased then the time required for the attacker to be in contact with the chip is also prolonged [16]. However, as this measure may seem to avoid skimming attack it does not resolve the problem of eavesdropping.

On the other hand, PACE enables PIN-protection to the data present on the identity document (eID card in our case) and it is a password authenticated Diffie-Hellman key agreement protocol that provides secure communication and explicit password-based authentication of the card chip and the terminal. PACE ensures that the contactless RF chip in the electronic ID card cannot be read without direct access and the data exchanged with the reading device is encrypted before getting transmitted. This protocol establishes Secure Messaging between a card chip and a terminal based on weak (short) passwords. Although the entropy of the password used to authenticate the local terminal can be very low (6-digit), the security provided by PACE is high because strong session keys resulting from PACE are independent of the strength of the password [2].

The transformation of a short password (PIN code) into a strong session key in PACE involves a *mapping function*, which is used to map a random number to parameters used for asymmetric cryptography. Two mapping alternatives are currently defined [16]:

- *Generic Mapping*: It is based on the generic group operations. This can be generically adapted to all asymmetric cryptography systems and is easy to implement on smart cards.

- *Integrated Mapping*: In this type of mapping, a random number is directly integrated in the parameters used for asymmetric cryptography. While this is easy to implement for standard cryptography, it requires more sophisticated algorithms for elliptic curve cryptography.

Furthermore, the cryptographic algorithms used by PACE for key agreement (KA), encryption (Enc) and message authentication (MAC) functions are Elliptic Curve Key Agreement (ECDH to be precise), AES 128 CBC-Mode and AES 128 CMAC respectively.

In the case of an offline authentication using nPA, the user is authenticated to the service provider's terminal by the initiation of PACE protocol as she enters

her PIN code. Then, a secure communication channel is established between the card and the service provider's terminal by the PACE session keys.

In the case of an online authentication, PACE is used for PIN-sharing and to create a secure channel only between the eID card and the local card reader on the user's side. The TLS channel established by the Ausweissapp with the eID-Server secures the communication between the eID-Client (the client application Ausweisapp which is further connected to the card reader and the card) and the eID-Server until the chip authentication session keys are established.

The PACE protocol between the card chip C and the terminal T is summarized in the following steps and also illustrated in the Figure 2.5.

1. C randomly and uniformly chooses a nonce s, encrypts the nonce to $z = E(K_\pi, s)$ where K_π=KDF(π) is derived from the shared password π; then the chip sends this ciphertext z to the terminal T along with its static domain parameters $Dparam_C$.

2. T recovers the plaintext $s = D(K_\pi, z)$ with the help of the shared password π.

3. Now, both C and T perform the following steps:

   (a) They compute the ephemeral domain parameters $EphDparam_C = Map(Dparam_C, s)$.

   (b) They perform an anonymous Diffie-Hellman key agreement based on the ephemeral domain parameters and generate the shared secret K like this:
   $K = \mathrm{KA}(\mathrm{SK_{Ce}}, \mathrm{PK_{Te}}, \mathrm{EphDparam}_C) = \mathrm{KA}(\mathrm{SK_{Te}}, \mathrm{PK_{Ce}}, \mathrm{EphDparam}_C)$.
   During this key agreement both the chip and the terminal must check if $\mathrm{PK_{Ce}} \neq \mathrm{PK_{Te}}$.

   (c) They derive the session keys: $\mathbf{K_{MAC}} = \mathbf{KDF_{MAC}}(\boldsymbol{K})$ and $\mathbf{K_{Enc}} = \mathbf{KDF_{Enc}}(\boldsymbol{K})$.

   (d) They exchange and verify the authentication token $\mathrm{AT}_T = \mathrm{MAC}(\mathrm{K_{MAC}}, \mathrm{PK_{Ce}})$ and $\mathrm{AT}_C = \mathrm{MAC}(\mathrm{K_{MAC}}, \mathrm{PK_{Te}})$.

The chip must allow only one execution of the PACE protocol within the same session. If PACE was successfully performed then the chip has verified the used password (i.e the eID PIN entered by the card owner on the local terminal is verified). Secure Messaging is started using the derived session keys $\mathrm{K_{MAC}}$ and $\mathrm{K_{Enc}}$.

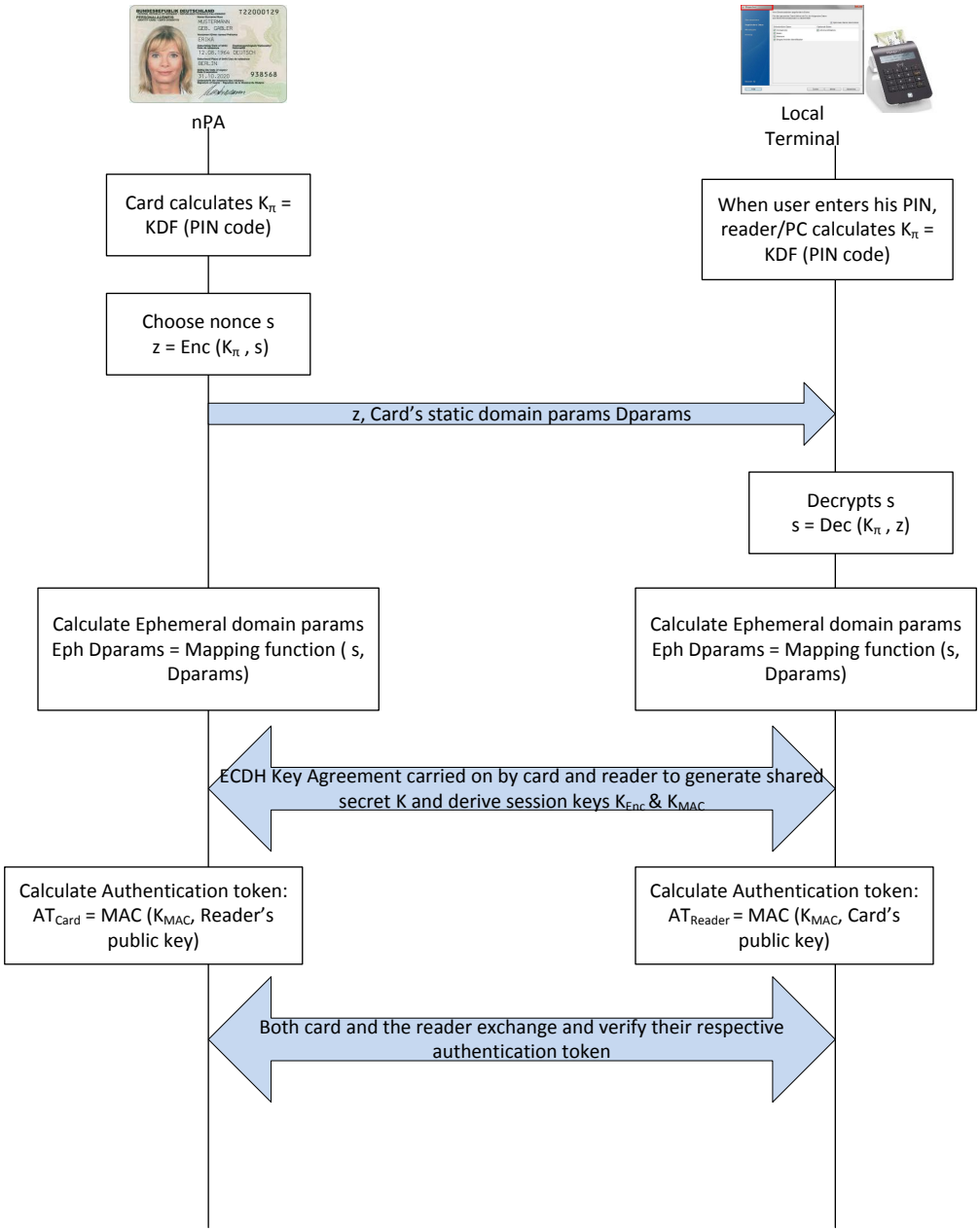

Figure 2.5: PACE protocol as used in nPA

2.4.2 Terminal Authentication (TA)

The Terminal Authentication protocol is a two move challenge-response protocol that provides an explicit unilateral authentication of the terminal. It is meant for the service provider terminal to prove to the eID card chip that it is entitled

to access the data on the chip. A service provider holds an authorization certificate encoding its public key and its access policy. The RF chip in the identity document is designed so that it enables reading of specific data only when the reader device (endpoint at an SP) can demonstrate an explicit read authorization for these specific data (e.g. only date of birth). The Country Verifying Certificate Authority certificate (CVCA certificate) is stored on the RF chip to verify this authorization. This certificate forms the root of the Country Verifier Public Key Infrastructure (CV-PKI), a hierarchy of authorization certificates for reading sensitive data from identity documents [1]. The TA authenticates the ephemeral public key of the terminal $\mathrm{PK_{Te}}$ that will be used to set up Secure messaging with Chip Authentication in the next phase. As the terminal might access sensitive data, further communication must be protected and the chip of the card must also bind the terminal access rights to the secure messaging session established by $\mathrm{PK_{Te}}$. In case of nPA, PACE is used before TA and the chip's identifier ID_C is computed using the chip's ephemeral PACE public key $\mathrm{PK_{Ce}}$. All messages must be transmitted with Secure Messaging in Encrypt-then-Authenticate mode using the session keys derived from PACE.

The TA between Chip C and the terminal T consists of the following steps [2] and it is also depicted in the Figure 2.4

1. T sends a certificate chain to C. The chain starts with a certificate that is verifiable with the CVCA public key stored on the chip and ends with the Terminal Certificate.

2. C verifies the certificates and extracts the terminal's static public key $\mathrm{PK_T}$.

3. T generates an ephemeral Diffie-Hellman key pair ($\mathrm{SK_{Te}}$, $\mathrm{PK_{Te}}$, Dparam_C) and sends the compressed ephemeral public key $\mathrm{Comp}(\mathrm{PK_{Te}})$ to the chip C. T may also send some auxiliary data AData_T to C.

4. C randomly chooses a challenge r_C and sends it to T.

5. T responds with the signature s_T calculated as shown below:
   $s_T = \mathbf{Sign(SK_{Te},\ ID}_C \parallel \boldsymbol{r}_C \parallel \mathbf{Comp(PK_{Te})} \parallel \mathbf{AData}_T\mathbf{)}$

6. C checks if $\mathbf{Verify}(\mathrm{PK_{Te}}, s_T, \mathrm{ID}_C \parallel r_C \parallel \mathrm{Comp}(\mathrm{PK_{Te}}) \parallel \mathrm{AData}_T) = \mathit{true}$.

The chip must allow only one execution of TA per session and if TA was successfully performed, the chip grants access to the stored sensitive data according to the effective authorization of the authenticated terminal. The terminal's access rights are however restricted to the secure messaging established by $\mathrm{PK_{Te}}$ and the chip will compare the value of $\mathrm{Comp}(\mathrm{PK_{Te}})$ received by the terminal as a part of

TA with the value of Comp(PK_{Te}) that will be calculated by the terminal during the Chip Authentication.

2.4.3 Passive Authentication (PA)

The purpose of Passive Authentication (PA) is to validate the authenticity and integrity of the data on the RF chip of the identity document. The data stored on the chip is digitally signed with a document signing certificate which in turn signed with a Country Signing Certificate Authority certificate (CSCA certificate)[12] that is available to an officially authorized ID card manufacturer. When an identity document is read, PA verifies the signature of the data stored on the RF chip and traces it back to the CSCA certificate thus verifying if the integrity of the data on the chip is intact and is indeed signed by the authorized ID manufacturer [1].

A more technical decription of PA is given below (For this, we have referred to the BSI TR-03127 [17]):

The EF.CardAccess file stored on the eID card consists of informational fields such as PACEInfo, CAInfo, CADomainParameterInfo, PrivilegedTerminalInfo, TAInfo and CardInfoLocator. These security information is collectively contained in a Security Object. This file is read by the reader terminal before the execution of PACE protocol as it is *always* readable by any terminal. On the other hand, the EF.CardSecurity file consists of CA PublicKeyInfo, RestrictedIdentificationInfo, DomainParams, signature over this entire Security Object data along with Document Signing (DS) authority's Certificate (which includes CSCA certificate) in addition to all the security information present in the EF.CardAccess file. Thus EF.CardSecurity file contains the secured contents of the Security Object and the read access right to this particular file is restricted to terminals that support PACE and TA and it can be read only after the execution of PACE and TA.

The terminal T conducts the following checks during PA:

- T compares the unsecured security information that was read from the EF.CardAccess file before PACE to the secured contents of the Security Object read from the EF.CardSecurity file after PACE+TA. T verifies if the corresponding information in both the files match and they have not been modified in any manner.
- T verifies the signature over the secured contents of the Security object present in the EF.CardSecurity file. As T already knows the public key of

[12]CSCA is operated by BSI

CSCA, it checks the validity of the DS certificate which includes CSCA certificate. Upon the successful verification of both signature and the certificate, the terminal T is assured of the authenticity and integrity of the data on the card is intact.

Passive Authentication must be performed in combination with Chip Authentication to confirm the genuineness of both the RF chip on the card as well as the data stored on this chip. If PA is performed after CA , the Document Security Object is used for PA (version 1). When PA is performed before CA, the Card Security Object (read from EF.CardSecurity file stored on the RF chip) or the Chip Security Object are used (Version 2). Only after a successful validation of the respective Security Object, the chip and its data may be considered genuine.

2.4.4 Chip Authentication (CA)

The Chip Authentication Protocol is an ephemeral-static Diffie-Hellman key agreement protocol that provides a secure communication and a unilateral authentication of the chip. The purpose of CA is to prove to the service provider that the eID card chip is an official chip issued by the German government (BSI), to confirm that it is not a forged or a cloned chip and to establish a secure connection between the chip and the service provider terminal in the case of an online authentication. The public key of the chip is signed by the card manufacturer during its process of generation; thus, the use of the signed key verifies the authenticity of the chip and at the same time, a strongly-encrypted and authenticated end-to-end channel is established between the chip and the service provider.

CA provides an explicit authentication of the chip by verifying the authentication token (smart card chip in this case) and implicit authentication of the stored data by performing the Secure Messaging using the new session keys [2]. In the present version of CA that is described in this section, Terminal Authentication (TA) is performed before CA where the terminal's ephemeral public key pair ($SK_{Te'}$, $PK_{Te'}$, $Dparam_C$) is generated as a part of TA (This ephemeral key pair is different from the pair used in PACE). The following steps are performed by the chip C and the terminal T during CA and CA is also illustrated in the Figure 2.4

1. Chip C sends its static Diffie-Hellman public key PK_C and the domain parameters $Dparam_C$ to the terminal T.
2. T sends its ephemeral public key $PK_{Te'}$ to C.
3. C computes the terminal's compressed ephemeral public key $Comp(PK_{Te'})$ and compares this to the compressed public key received in Terminal Authentication.

4. Both C and T compute the following:
   The shared secret $K = \mathrm{KA}(\mathrm{SK_C}, \mathrm{PK_{Te'}}, \mathrm{Dparam}_C) = \mathrm{KA}(\mathrm{SK_{Te'}}, \mathrm{PK_C}, \mathrm{Dparam}_C)$

5. C randomly chooses a nonce r_C and derives the session keys from K and r_C for Secure messaging :
   $\mathbf{K_{MAC}} = \mathbf{KDF_{MAC}}(\boldsymbol{K}, \boldsymbol{r_C})$ and $\mathbf{K_{Enc}} = \mathbf{KDF_{Enc}}(\boldsymbol{K}, \boldsymbol{r_C})$. C then computes an authentication token $\mathrm{AT}_C = \mathrm{MAC}(\mathrm{K_{MAC}}, \mathrm{PK_{Te'}})$ and sends r_C and AT_C to the terminal T.

6. T derives session keys for Secure Messaging from K and r_C:
   $\mathbf{K_{MAC}} = \mathbf{KDF_{MAC}}(\boldsymbol{K}, \boldsymbol{r_C})$ and $\mathbf{K_{Enc}} = \mathbf{KDF_{Enc}}(\boldsymbol{K}, \boldsymbol{r_C})$. Then, T verifies the authentication token AT_C.

If Chip Authentication was successfully performed, Secure Messaging is restarted using the derived session keys $\mathrm{K_{MAC}}$ and $\mathrm{K_{Enc}}$. Otherwise, Secure Messaging is continued using the session keys that were previously established during the PACE protocol.

One of the design principles in the German eID card was the non-usage of unique identifiers for the ID card in order to preserve the citizen's privacy to an extent and this is the reason for the Diffie Hellman key pairs not being unique for a chip [11]. For privacy reasons, the secret private key $\mathrm{SK_C}$ is shared with a batch of other cards to build anonymity sets so that anonymity only holds with respect to that set. Sharing a key is necessary, because otherwise a particular card could be traced using the corresponding public key that is transmitted to the terminal at the start of the chip authentication phase. Moreover, if each ID card were equipped with a unique chip authentication key, a service provider might gain a unique identifier as a side effect of the protocols. Therefore a batch of cards share the same secret chip authentication key, making them indistinguishable at the protocol level. We analyze the adverse side effects of this shared key approach as one of the main topics in Section 2.8.1.

2.5 Restricted Identification - Pseudonym feature in nPA

Restricted Identification (using pseudonyms) is a special option that is provided by the German eID where an eID cardholder can identify to an online service under a pseudonym. BSI proposed Restricted Identification (RI) protocol for the purpose of an eID cardholder's authentication over the Internet where the cardholder does not get directly recognized by her original name and data. The RI protocol is supposed to be executed after a secure trusted channel has been established through

secure messaging between the eID card and the service provider. Although the chip on the card is authenticated to the service provider during the chip authentication phase, the card's certified public key is shared among a large group of eID cards, thus the card is not uniquely identified. But the pseudonyms has to be unique for that service; so RI protocol generates the pseudonyms by combining the card owner's identity and the service provider's public domain value such that each pseudonym is unique to an eID card and a service. In the case of nPA, a user can acquire a single pseudonym for any sector like eHealthcare and the user can access services within this sector by identifying herself under that pseudonym. These pseudonyms are issued once and their lifetime ends when the card gets revoked due to expiry or the card gets lost. The pseudonyms have to satisfy the following requirements [18] when they are used for an online authentication:

- The pseudonyms must allow the service provider to recognize an eID card without requiring the cardholder's personal data. The cardholder should be able to maintain a state and history for a specific service. Hence, a terminal within a domain (e.g. which is associated to a service provider), is able to profile cards (i.e users). This property is called domain-specific linkability. Moreover, we require that the pseudonym under which a chip card authenticates to a terminal is unique in order to prevent Sybil attacks[13] and identity transfer.

- However, the pseudonyms should only be linkable within a domain. Two service providers (associated to different domains) should be unable to link interactions of the same user. We call this property cross-domain anonymity.

- For privacy reasons, full disclosure of the user's identity is unacceptable. A cardholder should not reveal more information than necessary for the specific service. For instance, an age verification should only reveal the age or even merely that the owner is above 18 years old. As such, the authentication process should be privacy-friendly.

BSI addressed all of the aforementioned requirements when they designed the Restricted Identification (RI) protocol for the German electronic identity card. The Restricted Identification Protocol is a static Diffie-Hellman key agreement protocol that generates a sector-specific identifier of the eID card chip. The pseudonyms are derived using only the unique secret key of a card and the certified domain-specific public key of a service provider.

In the **Restricted Identification protocol** the service provider terminal 'T' and the chip of the eID card 'C' perform the following steps:

[13] In Sybil attacks, a malicious party can illicitly acquire many pseudonyms. In our scenario, a malicious card could identify to the same service provider terminal using multiple pseudonyms.

1. T sends its certified public domain key PK_{Sector} and certain domain parameters *dparam* (distinct from the domain specific parameters used for PACE and EAC which were provided by the chip) to C.

2. C is able to verify the validity of the public domain key PK_{Sector} using public key of the CA, PK_{CA} that was a part of Terminal certificate exchanged during EAC.

3. The chip C computes the domain-specific pseudonym DS_{nym} as: $\boldsymbol{DS_{nym} = \mathbf{Hash(DH((PK_{Sector}, dparam), SK_{ID}))}}$ where Hash of Diffie-Hellman value is calculated by combining the chip's secret key SK_{ID} with the service provider terminal's (PK_{Sector}, dparam). Then, C sends DS_{nym} over to T.

4. T checks whether the received domain-specific pseudonym DS_{nym} is listed in the blacklist. If yes, T rejects and C is not identified. Otherwise, DS_{nym} is authenticated.

During the chip authentication phase, if the chip of the eID card sends its information to the service providers through its public key, then the service providers from different domains could link exactly this key and, consequently, link the pseudonyms, as well. This implies that cross-domain anonymity of RI is lost when applied after the chip authentication protocol. For this reason, the BSI proposed to share the public key between a group of cards. That is, several chip cards share the same public key, and, thus, the chip cards identity is hidden in this group of chip cards. Certainly, the size of these groups needs to be large enough to provide enough obfuscation and make the probability of linking the correct chip cards negligibly small.

Issues with nPA pseudonyms are listed below:

- The nPA proposes one pseudonym for a sector, meaning many services in that service sector identify the user under the same pseudonym and these services can still map the user across services in the same domain and link the transactions made by that user. This compromises the privacy to an extent and it may not be acceptable when the information can be considered sensitive and private for instance, health sector. But this issue can be addressed by suitably deciding on the granularity of the services.

- As the pseudonyms are generated using the secret key SK_{ID} of the card, if the eID card is lost then the pseudonym has to be revoked and a new pseudonym corresponding to a new secret key (of the new card) must be reissued. This results in the user losing all the previous transaction history

w.r.t the service provider. No identical pseudonyms can be reissued in such cases.

- It is not possible to use the same pseudonym on different authentication devices for instance, nPA authentication via a pseudonym on mobile phone or tablet as the pseudonyms are closely bound to the smartcard due to the fact that its secret key is being used for the pseudonym generation. This curbs the flexibility of using pseudonyms.

2.6 The eSign function in nPA

The German eID can also be used to digitally sign the documents with a Qualified Electronic Signature (QES). QES is a digital equivalent to a legally binding, hand-written signature according to the German Digital Signature Act [19]. The chip of the new ID card is designed to be a signature card in the sense of the German Digital Signature Act, i.e. citizen can use this card to load a qualified digital certificate and to sign the electronic documents in the usual way [11]; for example, an eID card can be used for signing the electronic contracts such as power of attorney or certain lease agreements. Unlike the online eID function (eID) with which the users can quasi-present themselves ("It's me!"), the QES is used to declare that the user agrees to a certain circumstance ("I agree to this")[14]. The eID authentication function just provides a 'snap-shot' of authentication to both the parties involved namely, the eID cardholder and the service provider. With this, they cannot prove the other's identity to another third party whereas a digital signature constitutes of such a proof that can be presented to a court or in any administrative proceedings if necessary. Just like the eID function, it is optional for the users whether or not to enable the eSign function on the eID card.

In order to use this signature function, the online ID function must be activated and the user requires an additional signature PIN along with her individual eID PIN. The eID PIN, PUK and eSign PIN are provided to the user at the time of issuance of eID card. The steps follows to exercise this eSign capability of nPA are as follows:

- *Key generation*: The eID card owner must create a signature key pair (pk_{sign}, sk_{sign}) and the signature application permits the creation of one key pair for the purpose of QES creation. The private key sk_{sign} is kept secret by the user, whereas the corresponding public key pk_{sign} is made public, for example distributed to all the registered verifiers (e.g. service providers).

[14] http://www.epractice.eu/files/eIDServicePocketGuide2011.pdf

- The user needs to *obtain a signature certificate* for this key pair from an accredited Certification Service Provider (CSP) such as Bundesdruckerei's trust center or D-TRUST. For this purpose, the eID card owner can identify herself to the CSP while requesting for a qualified certificate online. As the signature key pair is generated on the eID card, public key pk_{sign} is sent to the CSP and finally the certificate is loaded online onto the eID card. Then the card becomes ready to securely sign documents.

- *Signature generation*: The eID card owner who is now in possession of the certificate for the key pair (pk_{sign}, sk_{sign}) can produce her own digital signature σ on some message m. During the signing process, the user has to place her eID card on the reader device and enter the signature PIN. While a basic card reading device is sufficient for using the online eID function, an advanced card reader is required for the eSign functionality.

- *Signature verification*: Any authentic signature terminal belonging to a registered verifier (e.g. service provider) can execute a verification algorithm and check the validity of a signature σ on the given message m using the public key pk*sign* of the purported signer. This algorithm decides whether σ is valid or not.

2.7 Revocation method in nPA

To prevent the abuse of a stolen or a lost identity card, an eID cardholder must be able to block or cancel the card via a revocation management scheme [20]. If the eID card chips have a dedicated public-private key pair, then revocation is simple because the cards can be cancelled by means of a chip-specific public key that can be compared with a global revocation list. But such chip-specific feature is unique for a person and this leads to a direct identification or tracking of the eID cardholder thereby undermining the data protection and privacy friendly design of the nPA's eID function. For example, an online age-restricted service that requires only the proof of age must not be able to use a unique revocation attribute to cross-reference these data with a service that receives other attributes of the user such as name, address etc., from the identify document. This aspect is important especially when it comes to the restricted identification (Pseudonym) function of nPA as it thrives to achieve cross-domain unlinkability to preserve the privacy of a user. To avoid such conflicts, nPA uses service-specific revocation lists. This means that every identity card transmits a service-specific and card-specific revocation attribute to the service provider during the electronic identification process, which the provider then checks against his individual, i.e. service-specific revoca-

tion list [1].

If a service provider supports nPA's eID function, then a service-specific revocation list is generated from a global revocation list. The chip on the eID card calculates a card-specific and service-specific revocation token and sends it to the service provider during an authentication. This token can then be compared with the entries in service provider's revocation list in order to identify revoked ID cards. The procedure followed in this revocation scheme is describes in the following steps:

- *Initialization of the Revocation Service*: The revocation service generates a key pair and publishes its public key referred to as the revocation sector public key $\mathrm{PK}_{RevSector}$.

- *Initialization of a service provider*: The eID CA generates a revocation key pair for its registered service providers; the public key referred to as PK_{Sector} is calculated from the chosen private key SK_{Sector}[15] and the revocation sector public key $\mathrm{PK}_{RevSector}$. The service provider then receives a certificate containing its sector public key PK_{Sector}.

- The *ID card manufacturer* generates a revocation key, a revocation password and a revocation code and sends them to the revocation service. The revocation key is the public key of the key pair generated during the production of the ID card (PK_{ID} of chip authentication) and for security reasons, its length is 256 bits; the corresponding private key (SK_{ID}) is stored securely on the card chip and used for generating the revocation token. The revocation password is a conventional password that is randomly chosen during the card production, sent to the municipality where it is stored in a database; this password cannot be changed and it is sent to the eID cardholder along with the eID PIN code for the eID card. The revocation code is a cryptographic hash value of a concatenation of the cardholder's date of birth, surname, first name and revocation password; this code is generated during the card production and sent to the revocation service along with the revocation key and to the municipality where it can be stored.

- In the case of loss or stealth of an eID card, the cardholder can initiate the revocation at the municipality, with the police or via the revocation hotline. Once the revocation code is transferred to the revocation service, it looks up the respective revocation key and projects it into the revocation sector, a mathematical operation that requires the revocation service's private key

[15]Note: SK_{Sector} is retained by the eID CA and not revealed to the service provider.

$SK_{RevSector}$. This so-called 'activated revocation key' is then distributed to all the eID certification authorities [20].

- The *eID Certification Authority (eID CA)* transforms this revocation key into a service provider specific revocation token by making use of the unique public key of the card PK_{ID} (used in Restricted Identification protocol) and service provider specific private key (SK_{Sector}).

- During the eID authentication, the eID card calculates the same revocation token in the same way as it calulates the service-provider specific pseudonym (Refer to Restricted Identification protocol in Section 2.5). This revocation token is contained in the service provider-specific revocation list; this enables the service provider to recognize that the eID card has been revoked.

- *Revocation of the service providers* i.e rights to read data from the chip should also be revocable. Usually CVCA certificates have a short validity (depending on the data that can be read from the chip from 2 upto 30 days) and a recall of such a certificate can be realized by the non-issuing of a new one for this service provider.

An overview of nPA revocation process is provided by the Figure 2.6. This figure is taken from the document on the privacy-friendly revocation method used in German national eID card by Bender et al. [20].

Advantages of nPA revocation management scheme are listed below:

- The use of the service provider-specific and the card-specific revocation tokens makes it impossible for a service provider to recognize an eID card that has already authenticated to another service provider thereby achieving privacy in the form of cross-domain unlinkability.

- The revocation service being the central institution in this scheme also cannot derive the service provider-specific and card-specific revocation tokens from the revocation keys (without the help from the service provider and the eID certification authority). Thus, it is impossible to track an eID card by making use of the revocation mechanism.

- The revocation of lost or stolen eID cards should be available all the time i.e 24 hours a day and 7 days a week even when the eID cardholder is away from home. This would require storing of all the personal information necessary for the identification of the cardholders along with the revocation keys in the revocation service's database. Such a generation of central database of the citizens is prohibited by the German law. The alternative solution that is

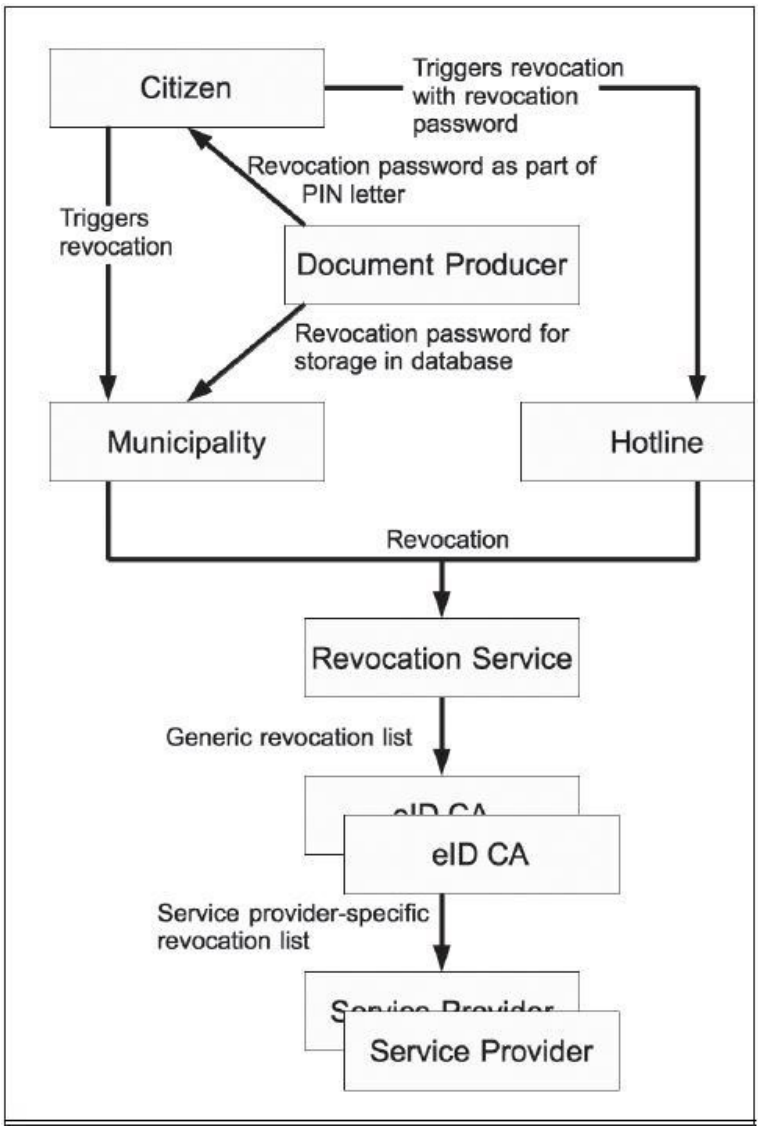

Figure 2.6: Overview of the nPA revocation process for a lost or a stolen eID card

adopted in nPA revocation scheme is that the revocation service's database stores only the revocation keys and their respective revocation code which is the cryptographic hash value over the concatenation of the cardholder's date of birth, surname, first name and revocation password. This privacy-friendly implementation of revocation management thus allows an effective revoking of ID cards without the need of a central register that contains the citizens' personal information.

Issues with nPA Revocation scheme:

- Revocation password is a randomly chosen conventional password sent to the municipality and to the user by the ID manufacturer and it cannot be changed by the user. This password is susceptible to hacks, for instance, dictionary attacks. Revocation code is in turn generated using the cardholder's date of birth, surname, first name (that are available on the card) and revocation password. So the entropy of the generated revocation code is low.

- Within the same domain of a service provider, the revocation token still makes tracking possible.

2.8 Limitations of German eID system

In this section, we discuss in detail about the two irreparable limitations of nPA arising from its design and also enumerate other disadvantages of nPA affecting security, privacy, flexibility and usability criteria.

2.8.1 Shared key concept in nPA that turns into a major limitation

Anonymity is defined as the inability to identify a subject within a set (the so-called anonymity set). Given a set of possible senders of a message, the real sender is anonymous within this set if she is not identifiable within this set [21]. The German eID is known to be the most privacy preserving solution till date and it seeks to achieve privacy by anonymity, by having shared key pair for the purpose of authenticating the eID cardholders. This is precisely the reason why *the chip authentication keys are not unique* in German eID approach.

During the Chip Authentication phase, the chip sends information about itself in form of a certified public key corresponding to its shared private key. As several chip cards share the same public key, a chip card's identity is hidden in a group of chip cards. Certainly, the size of these groups needs to be large enough to make the probability to link the correct chip cards negligibly small. Large groups for public keys provide sufficient obfuscation. But if one of the chips cards in this group is broken i.e an adversary manages to extract the secret key by breaking into the hardware of one card, then all identity cards which share the chip authentication key pair must be revoked. This is because the certified public key is the only information about an eID card which appears during the authentication. Hence, if all cards share the same public-private key pair, no card can authenticate after

suspicion of such leakage [18].

To avoid the risk of having to revoke a large batch of cards, privacy protection offered by shared key can still be achieved if the batch was small. In that case, we would have two options:

- The cards in the same batch could be distributed randomly over the country. This means that it is highly unlikely that two cards from the same batch (with the same key pair) will be owned by two people living in the same place. As a consequence, it is very likely that the public key of a card uniquely identifies a person, at least when she is using the card at or around her place of residence. This is usually a bigger concern when using the eID application in offline scenarios like buying cigarettes from a local vending machine than in online scenarios.

- Cards in the same batch are all issued to people that all live in the same place. But then the public key reveals the place of residence of the card holder.

So the privacy is compromised by having small group of shared keys whereas having a large group can maintain privacy but it runs the high risk of having to block all those cards which shared the key in case the secret key is compromised. Thus, the size of the group which share the same authentication key is a tradeoff between security and privacy.

According to Marian Harbach of BSI, chips that were produced within a period of three months would have got the same key pair for the chip authentication [11]. By October 31, 2011 (exactly one year from the introduction of nPA), 8 million new eID cards were issued[16]. This brings us to a figure of 2 million cards issued per 3 months in one year which implies that the same private key is shared by all those 2 million issued cards. The population of Germany as of year 2014 is close to 80 million[17]; if the German eID cards are issued to all the 80 million citizens by 2020 (10 years from its introduction), it would still mean 2 million cards are issued per 3 months and these cards will share the same chip authentication key pair.

The above facts and figures reveal the great extent of security risk in the case of nPA even if one of the issued eID cards is broken i.e. if the secret key of a single chip is extracted. If this happens, millions of eID cards have to be blocked;

[16] http://www.personalausweisportal.de/SharedDocs/Pressemitteilungen/DE/2011/Jahrestag.html
[17] http://worldpopulationreview.com/countries/germany-population/

blacklisting of cards becomes a reasonably big issue. In nPA, the entire security of the system is bound to the security and the tamper resistance of the chip. If the card hardware is manipulated and the private key or the secret stored on the chip of the card is acquired by the attacker, identity of the cardholder could be taken over, data integrity can be compromised etc., Hardware-based attacks (e.g. side-channel analysis or tampering bits on the chip by light attacks) are much more effective than cryptographic attacks. This aspect heightens the security risk of having a shared private key for a batch of eID cards. While thinking from an attacker's perspective with a certain attack potential, it is worth investing considerably large amount of time and resources in breaking one chip in nPA as it gives the attacker the chance to almost bring down the entire eID system by rendering millions of eID cards useless right after cracking just one chip.

2.8.2 Total dependency on the authenticity of chip and its operations - Individual attributes on eID card remains unsigned

Another important issue with the nPA that threatens the security of the eID authentication function is that the *eID data remain unsigned* [15]. The entire security and trust is tied to the authenticity and tamper-resistance of the eID card chip. The individual attributes on it and the results of eID special functions like age verification are assumed to be accurate and authentic. It means that the attributes/data groups that are transmitted after the selective disclosure by the user does not contain any signature in order to verify if it is indeed the data that was originally issued by BSI issuing authorities and sent by a legitimate eID cardholder. For instance, a service provider requests the eID cardholder to prove that she is over 18 years of age. Then the card chip runs its *age verification function* and replies *yes/no* to the service provider. There is no way for the service provider to check if the function gave an accurate result or if the eID card is not being mishandled by an adversary who is impersonating the legitimate eID card owner. One important point to be noted here is that in the passport application (ePass function of nPA) on the same card, the data groups are signed. The data that is specific just for the eID functionality is not signed.
The chip authentication establishes a session key between the eID-Server and the card resulting in a trusted channel between them. An access control policy in the card is bound to this channel, and the channel implicitly authenticates data sent through it. To prevent the service providers from proving to others that an eID record is authentic, there is no trusted party in the system that would sign the eID data [17]. Only the context of the EAC protocols run and the secure channel thus established assure the eID-Server of the authenticity of the eID data. Outside this

context, there is no way to verify the origin of eID data [15]. If the private key (SK_C) used for chip authentication is compromised, an attacker can create a card that can send arbitrary data that will be accepted by the server at face value. As a consequence, the attacker can impersonate an arbitrary person.

2.8.3 Other Security and Privacy issues

Other limitations of nPA in terms of security and privacy are listed below.

1. Shortly after the initial rollout of eID in November, 2010, some attacks were reported by the media[18] such as possibility of DNS manipulation on Ausweis-app, attack capturing the PIN when using a basic reader (one without the PIN-pad) and typing the PIN on an on-screen keyboard in the AusweisApp.

2. Dietrich et al. in their work [22] describe the Man-in-the-middle (MITM) attack against the intermediate network channels (TLS channels) that are subject to eID online authentication. The three TLS channels were between the web browser and other components namely, eID-Client, eID web service and eID-Server. This is the case with the basic card readers that were used during the initial roll out of the German eID card. Furthermore, according to [23], one of the three non-standard TLS channels used in nPA is between the service provider's web service and the user's browser and the eID-Client uses its own TLS connection to the eID-Server. So, if the connection is lost then there is no way to securely connect the eID-Client's TLS connection with any connection established by the browser.

3. Chip authentication during the authentication phase does not authenticate the cardholder but only shows, that the chip is an official ID card; thus has no security effect as an adversary could be in the possession of an official eID card [11].

4. If the eID-Server is deployed at a different location or domain than the eService and the handling of the eID-Server is in the hands of some third party, then there is a serious privacy risk. If the third party and hence the eID-Server is serving more than one service provider for the purpose of eID online authentication and the attributes that are revealed by the users on various authentication sessions are identifying, then the third party learns the visiting patterns of users over a large set of service providers (or relying parties).

[18] http://www.heise.de/newsticker/meldung/Neuer-Personalausweis-AusweisApp-mit-Luecken-2-Update-1133376.html

2.8.4 Usability and scalability issues

In spite of its advancements, privacy features and support from the federal government, nPA has usability issues that has influenced its wide adoption adversely. Some of them are listed below:

1. The nPA is a German federal eID where the data groups or the attributes stored in the eID card are predefined and issued by the government (BSI). Other private business sectors are not allowed to issue their own credentials/ attributes and load them onto the eID card. This poses as an impediment to the scalability and flexibility of the system; for instance, the use-case scenarios concerning the private eBusiness use-cases such as issuing Loyalty cards, monthly subscriptions etc., are not possible with nPA.

2. For the services that use the pseudonyms, nPA currently does not guarantee the issuance of identical pseudonyms on a new card if the card has been stolen or expired. This could be an issue for the citizen to identify herself to the previously accessed online services or to prove that her transactions in the past (before the card was stolen) was actually done by her.

3. According to the study done in [13], the current need to have a dedicated card reader was also repeatedly mentioned as a barrier to adoption. People generally highly valued the comfort of using smartphones, tablets and laptops and objected to the idea of reducing that comfort for purely security related reasons.

4. The currently available eID-Client 'Ausweisapp' only supports authentication with the German ID card on selected PC-based platforms. Important features such as the support for electronic signature techniques, other smart cards, the Mac OS platform or mobile devices are still lacking. As of today it is not clear whether and when those features will be supported [24].

5. As per the reports [25] [26] , the common people who received the eID cards were mostly unclear about the eID setup procedure; user experiences also indicated that the interface between the card application on the PC i.e Ausweisapp, card and the card reader is messy (vague terse error messages, user-unfriendly change-PIN, reset-PIN functions).

6. History function to assist the citizens in keeping track of their transactions and checking them at a later time has not yet been developed on the German eID. It is planned for future development [27].

Chapter 3

Attribute-based credentials

The actual identity of an individual is required for granting access to any resource in a physical environment whereas in an open environment as the Internet, grant decisions cannot be made based on the paper credentials because the owner and the requester of a resource belong to different security domains that are controlled by different authorities unknown to each other. In such situations, digital credentials can be used to satisfy the access policies of an online resource owner or a service provider. These digital credentials are often the characteristics of the requester rather than her full identity; they are digitally signed assertions about the credential owner by the credential issuer. The decision to allow or deny access to a resource is based on the *attributes* in the requester's credentials, such as age, citizenship, employment, group membership, or credit status. This approach is called *attribute-based access control*. The identity management systems are usually much more than just providing authentication and single sign on, such systems allow a user to reliably prove certain characteristics (called attributes) about herself to others. Typically, these attributes are used by the businesses to make a business decision during a transaction between the user and the business. A common example is verifying the age of a customer before selling alcohol or cigarettes. But more complex examples are possible, like verifying whether a user is a legitimate representative of a company with a certain creditworthiness or verifying that the user is a doctor that is allowed to prescribe a medicine of certain dose or quantity. In the realm of identity management, the business relying on these attributes to make an authorization decision is called a *relying party*. In generic terms, a relying party protects access to a resource and provides access to this resource depending on the credentials a user can show.

Trustworthy, yet privacy-preserving authentication is necessary to enable a long-term and lifelong privacy for users. Privacy-friendly architectures for identity

management do exist, for example, using attribute based credentials. In these systems, *credentials* are the secure containers that store the user attributes. They offer a solution that allows a strong authentication while the user may remain anonymous towards the relying party and without the identity provider learning about the websites visited or the services accessed by a user [28]. The people have to present an ID card or some form of credentials in many daily life instances such as opening a bank account or boarding a plane at the airport. These instances that are often handled offline do not enable the authority who checks the credentials to know where all the credential has been presented in the past, neither is the authority behind the counter equipped with the logs or the means to remember all the disclosed information about the user. But in the case of online transactions, the information not only lives forever but also creates the risks of revelation of extra unnecessary data and misuse of the data. For example, to get the student discount in a college canteen, one has to prove that one is enrolled in the college as a student. Only a binomial attribute '*Student*' saying *Yes or No* should be sufficient; it would be ridiculous if a student is asked to show her Passport or give her social security number for buying the discounted coffee. Authentication using attribute-based credentials aims exactly to do this i.e to check the authenticity and the authorization level and allow any transaction by requesting only the required attributes for that specific transaction.

The Attribute-based Credential (ABC) systems have been developed in the past to protect the privacy of attribute holder. Here, the users obtain the credentials from a credential issuer, that can vouch for the validity (for this user) of the attributes contained in the credential. The credential allows the user to prove the possession of an attribute to the relying party. Privacy is guaranteed by making the proof involving a credential unlinkable to other proofs involving the same credential or to the issuance of the credential. An important aspect of ABCs is that the credential issuer (analogously, the attribute provider) is not involved in the proof of possession of a credential. Moreover, using the technique of selective disclosure, the user can choose to reveal only a selection of the attributes contained in a credential.

Although the ABC technologies have been available for a long time, there has not been much adoption in the mainstream applications and the eID card implementations [3] due to the fact that the performance of ABCs on the smart cards (like eIDs) was poor and did not allow any practical deployment. Also, the complex concepts and the cryptographic mechanisms used in ABCs are hard to be understood by the non-specialists. But with time, ABCs and their implementations have evolved and now they have become attractive options to be integrated

in the privacy-preserving eID solutions. The EU-funded project ABC4Trust has been working on reducing the contraints faced by ABCs earlier when it came to practical implementations and a lot of progress has been made since in this direction. The recent research in Attribute based credentials [5] [6] has demonstrated the improved performance of the smartcard implementation of the ABC technologies. The IRMA (I Reveal My Attributes) project has been developed by Radboud University of Nijmegen in the Netherlands that demonstrates the applicability of ABCs on the smart cards. IRMA is being considered in this thesis as a suitable technology to be integrated into the eID systems in order to come up with a more secure and privacy-friendly eID authentication scheme and concepts of IRMA are explained in the upcoming sections.

3.1 I Reveal My Attributes (IRMA)

Two important technologies that make use of an ABC approach are Microsoft's U-Prove [29] and IBM's Identity Mixer (Idemix) [30] technologies. Idemix is an attribute-based credential system, developed at IBM Research in Zürich, that enables strong authentication and privacy at the same time. The Idemix technology is tightly built upon the concept of Camenisch-Lysyankaya signature scheme and its protocols [31]. It relies on the strong RSA assumption [32].

IRMA is a pilot project where a platform has been developed to support the attribute-based credentials on a smart card and it is a partial implementation of Idemix [5]. It is being carried out at the Radboud University of Nijmegen, The Netherlands. Conceptually, IRMA could also use the U-Prove technology for its implementation but at present, only the Idemix implementation is used in the pilot project. The IRMA project aims to demonstrate the applicability of attribute-based credentials (ABCs) in practice, on smart cards and it currently has implemented the privacy enhancing features of ABC such as selective disclosure of attributes using zero-knowledge protocols.
The main idea behind the IRMA card is that the information stored on the card can be read digitally from the card only if the cardholder gives an explicit permission to read a specified set of attributes (e.g. age, name etc.,). Currently, the IRMA card and the reader software is implemented and a small scale usability experiment[1] has been conducted with a couple of students at the Radboud University. At the university, students with IRMA cards can buy coffee at a lower price and also use some of the university printers for free. The IRMA card verification terminals (Android-based tablets with IRMA application running on them) have even been

[1]The pilot presentation can be found at https://www.irmacard.org/wp-content/uploads/2013/11/The-IRMA-pilot-18-11-2013.pdf

installed at the canteens in the University of Twente for the students with IRMA cards to buy coffee at a lower price. These IRMA terminals check the presence of a 'Student' credential issued by the University. These are small-scale projects but the final aim of the IRMA project is that the IRMA card would become an alternative for the current citizen identity cards[2]. As it can be seen in the Figure

Figure 3.1: Typical IRMA card

3.1, only the picture of the card owner and a serial number are printed on the IRMA card. The picture can be used to prove to the verifying party that the card actually belongs to the cardholder in the case of an offline authentication use case and the serial number is used to look up the owner of the card if the IRMA card is lost. This serial number is not stored anywhere on the chip of the card and thus cannot be used to link different transactions involving the card. In the case of online scenarios, only the digital proofs of knowledge are used for authentication purposes using IRMA card. To achieve higher privacy, IRMA allows the cardholder to reveal one or a set of selected attributes within a credential and also, a PIN code may be associated with an attribute that is considered as sensitive data (e.g. Social Security Number). The main advantage concerning privacy is that the IRMA cardholder does not have to reveal her uniquely identifying attributes like SSN in the situations that do not require it. For instance, IRMA card can be used to prove the possession of valid concert ticket or valid credentials to enter an office building by just revealing the "ticket" and "Is an employee" attributes rather than revealing all the attributes on the card. In such cases there is no need to reveal a unique identifying attribute and this attribute-based method prevents linking of different proofs and implicit profiling of the card holder. The main features and functions supported by IRMA system are described in detail in the coming sections.

[2]Attributen in plaats van Identiteiten by Bart Jacobs (https://www.irmacard.org/wp-content/uploads/2012/12/irma-intro-nl-dec2012.pdf)

3.1.1 Stakeholders in a IRMA-based identity management system

From a slightly higher level of abstraction, we can distinguish the following classes of **roles** applicable for an ABC-based (IRMA-based) Identity Management system:

1. **Scheme Authority (SA)** is the highest authority in the IRMA system as it governs the identity management scheme and is responsible to maintain the trust and value of the scheme. It is the entity that lays down the architecture of the system and rules for the other stakeholders of the IRMA system while governing their operations. It is responsible for ensuring the system is useful, economically viable, secure and respects privacy. The scheme authority, for instance, decides which providers (ID card manufacturers), credential issuers and relying parties are allowed to use the system (and revokes their access if they fail to comply with the rules). The Scheme Authority also initializes the smart cards that are going to be used for the IRMA system. These tasks could be split up into different organizations when the system is being used on full scale. Typically, relying parties need to authenticate and provide the appropriate certificates to the user in order to read out certain attributes. This *relying party authentication* (traditionally called terminal authentication) prevents relying parties from sneakily collecting all attributes of a user. The scheme authority can also take up the responsibility for issuing Authorization certificates to the relying parties for the purpose of relying party authentication.

2. **Users (IRMA cardholders)** that benefit from the functionality offered by the providers in the identity management scheme. The cardholders are the ones who have to prove to the verifiers that they possess the required set of properties or attributes in order to complete a transaction that the card has initiated. The verifiers or the relying parties can also take the place of users when they have to access certain service and prove their attributes to some other verifier.

3. **Credential Issuers** that compose the credentials as a secure container of attributes that hold for a user for a certain timespan. After making sure that the credential to be issued is destined to the current legitimate user and validating if the attributes inside the new credential actually hold for that user, the credential issuer signs the credential. This signature proves the authenticity of the issued credential and the attributes stored within it. If the issuer has to verify other credentials stored on the card in order to issue a new credential to the user, the issuance protocol may consist of multiple steps.

4. **Credential Verifiers** often known as the relying parties are the ones who require the proof of certain attributes of a user to complete the transaction. During the verification protocol, the verifier requests the user (or the IRMA card) for a set of attributes that need to be present on the card. The card answers with a proof of the credential along with the disclosed attributes. The verifier can then check the validity of the proof and if the credential issuer who has signed can be trusted. There can also be cases where the credential issuer and the verifier are the same.

5. **Providers (ID card manufacturers)** that offer identity management functionality i.e they provide the security tokens (typically smart cards) that the users can use to securely obtain, store, and use credentials.

6. **Revocation Authority** is a semi-trusted party in the IRMA system that is responsible for revoking credentials. Alternatively, the credential revocation could be merged with other responsibilities of the Scheme Authority.

7. **Pseudonym Authority** is a semi-trusted party in IRMA system that is responsible for generating and providing the pseudonym values as attributes to be a part of IRMA cards. The Scheme Authority can either take up this responsibility or just function as the receiver of pseudonym values from pseudonym authority and load it onto the IRMA cards.

3.1.2 Attributes & Credentials

An *attribute* is a characteristic or a property concerning a person. For a given person, his attributes could be the following:

- His name is 'Xxx'
- He is a student
- His age is over 18
- His address is 'abc...'
- His Social Security Number (SSN) is '12345'.

These attributes can either be *identifying* or *non-identifying* properties. In the above example, the attributes 'name', 'address', 'SSN' are identifying attributes as the person can be uniquely identified with the aid of such attributes. The attributes 'student' and 'age over 18' are non-identifying attributes as they do not uniquely identify a person; such properties can belong to other people as well. Attributes like 'age over 18' can be derived from the 'date of birth' attribute and

hence termed as *derived attribute*. Collectively, these attributes can constitute the identity of a person.

A set of attributes can be grouped into a cryptographic container known as *Credential*. Basically, the attributes related to a particular service or an activity are put in one credential.
The credentials are issued to the users upon request from the credentials issuing authorities after the verification of attribute statements. The credentials are issued and verified during the issuance procedure whereas the attributes contained in those credentials are disclosed or proved by the user to the respective service provider during the verification procedure. Figure 3.2 shows how a typical credential in IRMA looks like. In this figure, *s* denotes the user's secret key, *a1..a4* denote the attributes present in the credentials, *(A,e,v)* collectively denote the credential issuer's signature.

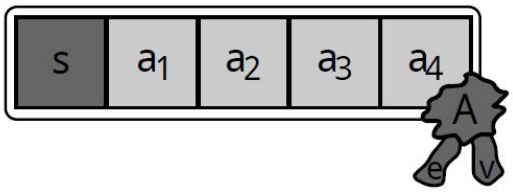

Figure 3.2: A visual representation of an Idemix-IRMA credential

The credential itself is never transmitted but it could be used to prove or convince the verifier that the cardholder's attributes fulfil certain properties without leaking further information. This is achieved in ABCs by using the cryptographic concepts of Zero-Knowledge proofs where a prover can convince a verifier that a mathematical statement holds without revealing further information.

The cryptographic nature of the credential-as-container concept includes the following four security aspects:

- The issuer's digital signature ensures authenticity: the credential originates from the issuer, and this issuer asserts that the attributes hold for the user.
- This signature also guarantees integrity: the attributes contained in the credential have not been altered since they were issued.
- A credential is non-transferable as it is bound to the secret key of the person involved in the issuance protocol. This secret key should be well protected, for instance via storage in the secure memory of a smart card with a PIN.

- A credential hides its content, so it does not reveal the attributes it contains. Data minimization principle is enforced by allowing the card owner to selectively disclose the attributes stored within a credential to the relying party while hiding the other attributes.

Furthermore, a credential protects the privacy of its owner through the following two cryptographic properties.

- Issuer unlinkability ensures that any information gathered during issuing cannot be used to link a verification of the credential to its issuance.
- Multi-show unlinkability guarantees that when a credential is verified multiple times, these sessions cannot be linked.

The privacy of users is protected by these unlinkability properties even if the credential issuer and all verifiers collude.

3.2 Cryptographic background of IRMA

In this section, we describe the main cryptographic concepts used in attribute-based credentials like a proof of knowledge, Schnorr's identification scheme, CL signature scheme that is used in Idemix. The building blocks of Zero knowledge protocols are used in the Idemix implementation of IRMA. We refer to the Cryptographic preliminaries chapter of the PhD thesis by Vullers [33] in addition to the original sources for Schnorr protocol and Camenisch-Lysyanskaya signature concepts.

3.2.1 Proof of Knowledge and Zero knowledge protocols

A *proof of knowledge* is a proof with which a user or a prover can convince a verifier of a given statement or of having a certain knowledge. For example, a prover can authenticate herself to the verifier by providing a proof that she knows the private key corresponding to the public key used by the verifier. This can be done by constructing a challenge-response series where the user signs or decrypts a challenge with the help of her private key.

A prover can prove his knowledge of a value without revealing the value or any additional information with a Zero Knowledge (ZK) protocol. Zero-Knowledge says that, no malicious verifier can extract any useful knowledge from the prover, no matter what he does. Moreover, the term *zero-knowledge* basically indicates that the user's information learnt by the verifier could have been generated by the

verifier himself without the assistance of the user. But when the user provides a zero-knowledge proof, the verifier becomes convinced of the fact that the user has the specified knowledge (e.g. private key).

A well-known example for a ZK protocol is Schnorr's identification protocol [34] which is a protocol for proving knowledge of a *discrete logarithm* and it is described below. All the zero-knowledge proofs in the Idemix library are implemented as a common three-move ZK protocol similar to Schnorr's protocol. The ZK protocols used in Idemix are made non-interactive using the Fiat-Shamir heuristic [35].

Schnorr's Identification protocol

Schnorr's identification scheme [36] is a simple three-way zero-knowledge scheme which proves the knowledge of a discrete logarithm x of a number y (mod n):

$$PK\{(x) : y = g^x \ mod \, n\} \tag{3.1}$$

where PK denotes the proof of knowledge, x is the discrete logarithm of y and g is the generator belonging to the cyclic group G_n with order n.

Figure 3.3 illustrates both the interactive and non-interactive proofs used in Schnorr's identification scheme.

The values that are public and hence known by both the prover and the verifier are: g, n, y. But the value of discrete logarithm of y is known only to the prover. In order to prove the knowledge of $x = log_g y$, the prover interacts with the verifier as follows:

- The prover commits herself to a random value r; therefore the first message $t = g^r$ is also called the *commitment*.
- The verifier replies with a challenge c chosen at random.
- After receiving c, the prover sends the third and last message (the response) $s = r + cx$.

The verifier accepts, if $g^s = ty^c$.
The security of Schnorr's identification scheme relies on the hardness of discrete logarithm problem.
In the interactive proof, the verifier can be sure in the last step that the prover knows the discrete logarithm of 'y' if it satisfies the condition $(t = g^s y^{-c})$ because,

$$t = g^s y^{(-c)} = g^{r+cx}(g^x)^{-c} = g^r \tag{3.2}$$

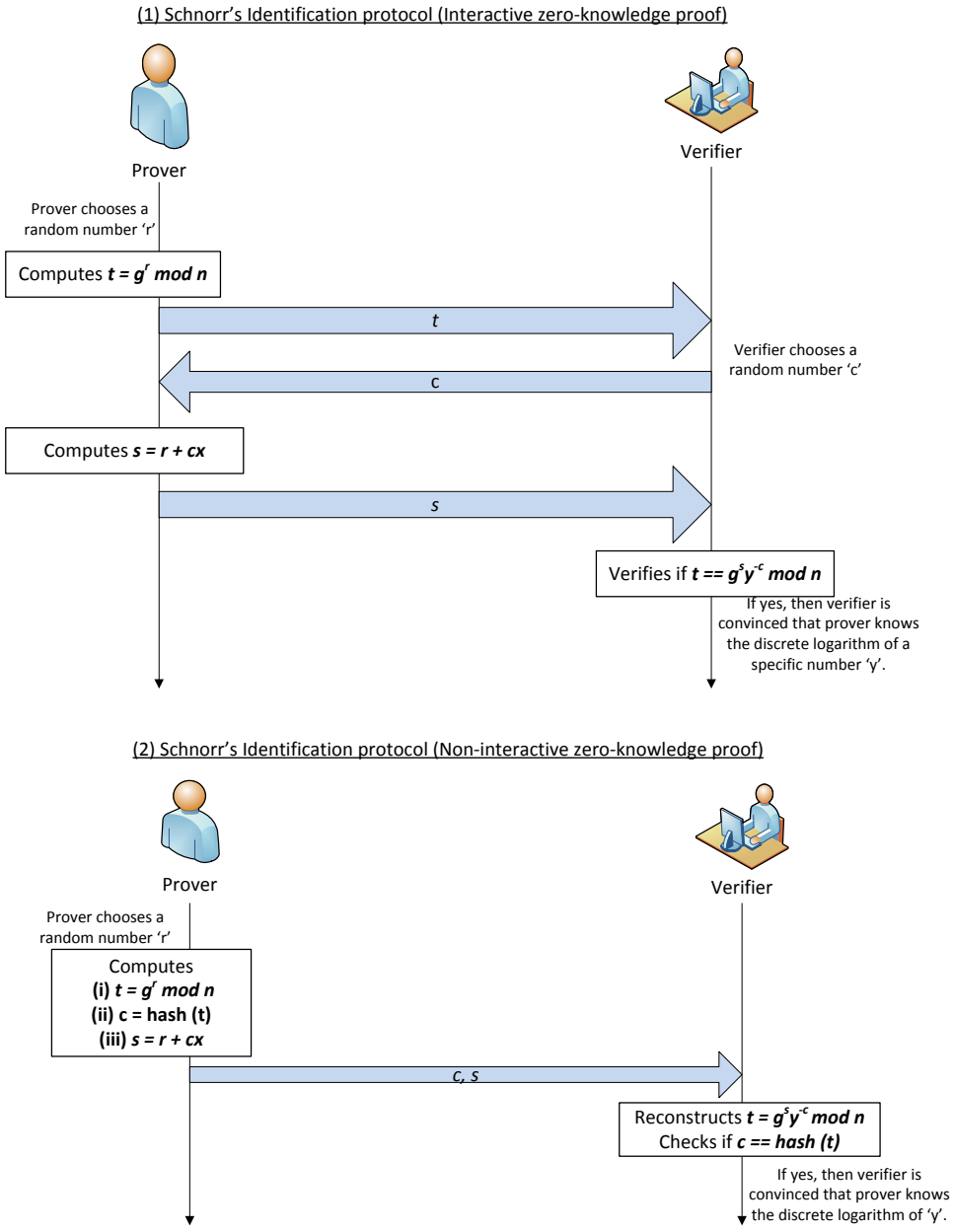

Figure 3.3: Schnorr's identification protocols

The verifier does not learn anything from such a conversation (t, c, s), since he could have computed such a triple himself by choosing c and s at random and computing $t = g^s.y^{-c} \ mod \, n$. This means that the zero-knowledge property holds

for this protocol.
Another important property is soundness, which guarantees that the user actually knows the the secret. Suppose that given a single commitment a the user is able to respond to two different challenges, hence generating two conversations (t, c, s) and (t, c', s') where $c \neq c'$. We will have $t = g^s.y^{-c} \, mod \, n$ and $t = g^{s'}.y^{-c'} \, mod \, n$, it follows that

$$g^{s'} = g^s.y^{-c'}.y^{c'} \, mod \, n \tag{3.3}$$

which implies that

$$g^{s'-s} = y^{c'-c} \, mod \, n \tag{3.4}$$

Hence, $y = g^{\frac{s'-s}{c'-c}} \, mod \, n$ which means that the user knows the private key x since, $x = \frac{s'-s}{c'-c} \, mod \, n$.

The non-interactive scheme reduces the number of rounds of information exchange between the prover and the verifier thus, saves time and bandwidth in a network environment. The non-interactive Schnorr's identification is achieved through making the challenge c a hash value of t (See Figure 3.3) The correctness of the non-interactive proof is shown as:

$$hash(t) = hash(g^s y^{(-c)}) = hash(g^{r+cx}(g^x)^{(-c)}) = hash(g^r) = c \tag{3.5}$$

3.2.2 Camenisch-Lysyanskaya scheme

The Camenisch-Lysyanskaya scheme [32], as used in the context of IRMA is described by Camenisch et al. in their work [37] and also in the specification of Idemix cryptographic library [38]. The cryptographic preliminaries like safe primes and special RSA modulus used in the CL-signature is first explained before we move on to explain the scheme itself.

Safe Prime: A safe prime x is a prime number of the form *(2y+1)* where y is also a prime number. In the context of this paper, we take p and q as safe primes that are of the form $(2p' + 1)$ and $(2q' + 1)$ respectively where p' and q' are also primes.

Special RSA Modulus: The special RSA modulus is an RSA modulus $n = pq$ with p and q are both safe primes. It can be seen that the size of the RSA group $\phi(n) = \phi(pq) = 4p'q'$.
If we consider the subgroup of quadratic residues modulo n, $QR_n \subseteq Z_n^*$; size of the subgroup is $\mid QR_n \mid = \frac{1}{4}(4p'q') = p'q'$.

CL signature: CL scheme uses a RSA based approach where the RSA modulus n is the public part of the key, whereas the private part consists of the primes p and q. This allows them to generate a fresh exponent e for each signature which will then become part of the signature. Since an attacker can now control both the signature and the exponent, solving the RSA problem becomes easier. Hence a stronger assumption is needed. This strong RSA assumption states that the probability that an attacker can solve the RSA problem is negligible, even when the attacker can chose the public exponent e.

In Schnorr's identification scheme, we can use the groups where the order of the subgroups are known to all parties but in the context of CL-signature used in Idemix, we consider the groups in which the order of the (sub)group is not known to all parties. This is because, in an attribute-based credential system, we would like only the issuer of the credential to know the order of the group and only he can sign the attributes within a credential ensuring the authenticity of the whole credential.
The CL scheme uses a variant of Schnorr's protocol with a single group and a modulus n (as explained in the section 3.2.1). This is the case in an RSA setting where the order of the group is only known by the party that knows the primes p and q (issuer in the ABC case). As a result, the user cannot perform the modular reduction using the order of the generator g when computing the response s (Refer Figure 3.3). This means that the response no longer hides the secret x as it is not distributed uniformly. Therefore the user must choose a significantly larger[3] random value u such that a is distributed statistically close to uniform over the subgroup generated by g and x is statistically hidden in the response s. Hence, this protocol is called *statistical zero-knowledge* [39].

With this background, we briefly explain the structure of a CL signature. A CL signature is a signing scheme which enables a server to sign the information provided by the users. In the context of ABC, each attribute inside a credential is considered as a message m. In order to sign a collection of messages $\{m_i\}_{i \in M}$, these m_i, they have to be aggregated into a single group element Q according to the following equation:

$$Q = \frac{Z}{S^v \cdot \prod_{i \in M} R_i^{m_i}} \mod n, \tag{3.6}$$

where v is a random number. The value v is used for blinding the messages (i.e.

[3]For instance, according to Vullers [33] the length of the random value u should be 80-bits longer than the combined lengths of the modulus n and the challenge n. This in contrast to just the length of the prime order q.

attributes) that have to remain hidden from the issuer (Refer subsection 3.2.3) and it is also used to randomize the signature (Refer subsection 3.2.4).

The actual signature generation process is similar to the RSA signature scheme. The steps in which CL-signature is created are listed below:

1 Generation of a random prime e which is used as the ephemeral RSA public key for this signature.

2 The RSA private key $d = e^{-1} \mod (p' \cdot q')$ corresponding to the public key e is computed. In IRMA (Idemix) case, only the credential issuer knows the values of p and q which he can use to invert the value of e. No other entity can invert e.

3 Finally, $A = Q^d \mod n$ is the RSA signature over the aggregated messages.

As a result the Camenisch-Lysyanskaya signature over the messages $\{m_i\}_{i \in M}$ is the triple (A, e, v).

In order to *verify* such a Camenisch-Lysyanskaya signature (A, e, v), the RSA signature over the aggregated messages has to be verified. That is, the verifier has to check the following equation:

$$A^e = \frac{Z}{S^v \cdot \prod_{i \in M} R_i^{m_i}} \mod n \tag{3.7}$$

If the verification result is true, then the signature is valid; otherwise the signature is invalid.

3.2.3 Blind signature scheme

Blind signatures are used in IRMA to ensure issuer unlinkability which means that any information gathered during the issuance cannot be used to link a verification of the credential to its issuance. The credential issuer issues the credential with his digital signature on the committed values. According to this scheme, the issuer being the signer has no information about the signed value and he also proves the knowledge of this signature. Even in a situation where the issuer himself becomes the verifier, the issuer will not be able to link the credentials that was issued to the IRMA cardholder by him.

This protocol hides the messages $m_i \in M_H$, where $M_H \subseteq M$, from the signer by generating a commitment to these values and blinding them. During this commitment phase these messages are aggregated into a single element U and hidden by the blinding value v'. Note that the remaining messages $\{m_i\}_{i \in M \setminus M_H}$,

that are not hidden during this phase are made known to the signer.
In order to prove to the signer that the user actually knows the hidden messages, the following proof of knowledge has to be carried out.

$$PK\{(\nu, \mu_{i \in M_H}) : U = S^{\nu} \cdot \prod_{i \in M_H} R_i^{\mu_i} \mod n\}$$

This proof not only proves that the user knows the hidden messages, but also that the value U has been constructed correctly by the user. This can be implemented as an interactive zero-knowledge protocol as described in the section 3.3.

3.2.4 Signature randomization

If the credential issuer's signature is unique and constant over multiple sessions, then, when the IRMA cardholder authenticates to a service provider, her transactions can be linked and the user can be eventually traced. The signature of the attribute issuer i.e the CL-Signature (A, e, v) is randomized to prevent linkability based on the CL-signature values A, e, v. The signature is randomized according to the following steps:

1 A random blinding value r_A is generated to randomize A.

2 Next, the values e and v are adjusted such that the signature remains valid. The adjusted values of e and v can be denoted as e' and v' respectively.

As this randomization operation effectively randomizes only A value, the values e' and v' need to be hidden by the zero-knowledge proof while revealing the signature to the service provider. This randomization process prevents any service provider from linking or gaining information based on the attribute issuer's signature about the user transactions in different domains or in the same domain. This allows the credential to be used multiple times while the user remaining anonymous thus, providing multi-show unlinkability.

3.3 IRMA card and Credential Issuance

Once the IRMA application is loaded onto the IRMA card by the Scheme Authority, the card has to be bound to the new owner of the card. This is done by issuing a basic credential (root credential) with the master secret and the most important personal information of the owner. The IRMA card issuance is technically similar to the credential issuance procedure; the only part where the issuance of this first basic credential is different from the credentials by other issuers is that the master secret needs to be generated on the card. This master secret is a *256 bit random key*. In the Idemix set-up, this master secret will be used for all the credentials on

the card.

When a new credential has to be created and issued to an IRMA card, this credential has to be bound to that specific card only; to ensure this, following steps are followed during the issuance protocol and Figure 3.4 depicts the credential issuance protocol in Idemix.

- Initialy, the user requests a credential from an authorized issuer and based on this request, the issuer decides on the attribute values for that credential. Further, in IRMA, the issuer sends his public key as a part of this initialization phase before the credential construction process begins.

- *Card commitment phase:* Firstly, the issuer generates a fresh *nonce* n_1 and sends it to the card in order to make sure that it receives a fresh commitment from the IRMA card and it is not being a victim of a replay attack in which case the issuer might be receiving copied commitments/responses from another session. Then, in order to convince the issuer that it is an authentic IRMA card with a user-bound secret s, the card generates a commitment U using the master secret key s. As the master secret key may never leave the card, the commitment scheme is used. The commitment U is computed from the secret key s by generating a random value v; it is calculated as,

$$U \leftarrow S^v.R^s \; mod \, n \tag{3.8}$$

  where *S, R and n* are part of the public key of the credential issuer which is already known to the IRMA card.
  The issuer is unable to calculate the value of the commitment U as he does not have the secret key s; so, a non-interactive proof of correctness P_U is computed. With the help of P_U, the issuer can verify the validity of U.

- *Issuer's signature phase*: The card generates a *nonce* n_2 to make sure that the signature is fresh. If the issuer has successfully verified the proof of the commitment P_U in the previous step, it signs the attributes that are part of the credential along with the commitment U and nonce n_2. The scheme used for this signature is the Camenisch Lysyanskaya (CL) signature scheme [32] in which the signature S has to be accompanied with a signature proof P_S. The card can validate the issuer's signature based on the proof P_S and the public key information that is sent by the issuer to the card during the initialization phase. The attributes are signed and thus the credential is successfully constructed and stored on the card.

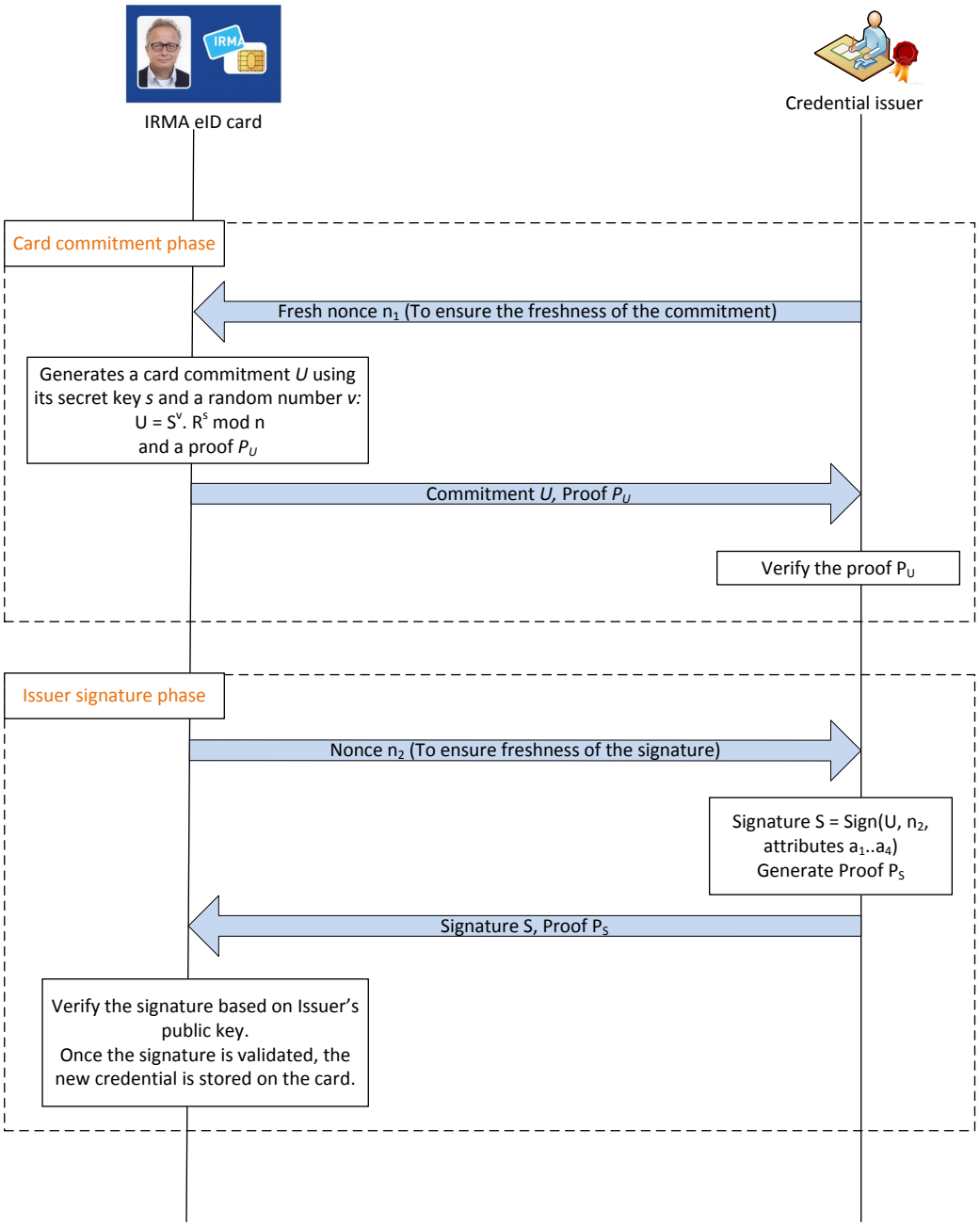

Figure 3.4: Overview of the credential issuance protocol in IRMA

3.4 Selective disclosure in IRMA

The concept of revealing only a selection of necessary attributes for completing a transaction is termed as *Selective disclosure* in IRMA and it fits perfectly within

the principle of privacy by data minimization. The verifier is obliged to state the required set of attributes and ask the user if she wants to use the IRMA card in order to prove that a certain attribute is present on the card. The user can check if the requested attributes are reasonable and absolutely necessary before cooperating to start the IRMA procedure of proving the presence of those attributes. IRMA uses the notion of a smart card with a PIN code which naturally creates higher alertness in users when deciding about these transactions.

In IRMA, the credential (selective disclosure) operations are performed in a group in which the Strong RSA problem is hard. A master secret key stored on the chip of the card is used in selective disclosure of attributes. All the credentials contain (1) the issuer's signature vouching the presence of the device master secret 's' which binds the credential to the device and (2) the expiry date attribute 'a_{exp}' that indicates the validity of the credential to the relying party when all/selected attributes within the credential are revealed. The credentials can be distinguished into two types: root credential and the regular credentials. A root credential can be made to contain special attributes such as the personal pseudonym attribute 'a_{nym}' and the revocation value of the card 'a_{rev}' whereas the regular credentials do not contain any such special values. The root credential is used when a pseudonym and/or revocation check is required, whereas regular credentials are used to reveal attributes. This can also be combined, either by including attributes in the root credential or by using additional credentials besides the root credential [40]. During the selective disclosure of attributes, expiry a_{exp} is always revealed so that validity of the credential can be verified but the values of 's', 'a_{nym}', 'a_{rev}' are never revealed. Any other attributes contained in a credential are denoted as a_1,...,a_l, where l is the number of attributes. The signature over the credential containing these attributes is the triplet *(A,e,v)*[4].

Selective disclosure and zero-knowledge proof :
When an online transaction is initiated in IRMA, a service provider requires certain attributes of the user to authenticate her and grant access to the service. For this purpose, the service provider sends a request to the user that contains an attribute disclosure selection D to indicate which attributes should be revealed and a fresh *nonce* to ensure a freshly generated proof. The operations carried out by the user or the IRMA cardholder during a selective disclosure proof are enumerated in the following steps:

1. If the pseudonym was requested by the service provider, then the card con-

[4]Camenisch-Lysyanskaya signature over the attributes is the triplet (A, e, v) where 'e' is the random prime used as the ephemeral RSA public key for this signature and 'v' is a random number and 'A' is the RSA signature over the aggregated set of attributes within a credential.

verts the service-provider identifier required for pseudonym generation (e.g. URI of SP) into a generator g_{nym} and then randomize the issuer signature *(A,e,v)* (Refer to the section 3.2.4 for signature randomization explanation).

2. Next, the card generates a zero-knowledge proof that proves the validity of the credential containing the requested attributes and also hides the undisclosed attributes $a_i \notin D$, e' and v' values. This proof binds the pseudonym and revocation values to the signature to ensure authentication and integrity of the generated values.
   The zero-knowledge proof is constructed in the following four steps:
   (a) A set of blinding values is generated to hide all the values.
   (b) Commit to these binding values by calculating the commitments.
   (c) These commitments are used to calculate challenges which are forwarded as input to the next step.
   (d) Calculation of the responses to the challenges. Finally the proof, challenges, responses and the revealed values are returned.

The service provider receives attributes $a_i \in D$, generated pseudonym ID_{nym} and revocation value ID_{rev}. The verification of the received values takes place as follows:

1. Reconstruct the commitments using the responses (and the challenge) sent to it by the card.
2. Construct the challenge using the commitments and match it with the received challenge.
3. If it matches then the proof is valid. Otherwise, reject all the values and abort the session.
4. Once the proof is validated, the time validity of these values can be verified by checking the a_{exp} attribute (As the proof is valid, we can be sure now that the attribute a_{exp} value is also valid).

3.5 Data minimization functions in IRMA

As the name suggests, the data minimization functions aim at revealing just the information needed to complete the transaction. This is one step ahead of selective disclosure as in the case of data minimization function, just the boolean values (yes/no or 0/1) can be stored corresponding to the attribute names for example, "age $\geq$ 18?" attribute; this will not even reveal the actual value of that attribute (actual age in this example).

Age verification is one of the most familiar data minimization function and it is implemented in IRMA in the following manner.
At the moment, there are predefined age groups as attributes in IRMA and simple reveal or hide zero-knowledge proofs are used. Age credentials look typically as shown in Table 3.1; for instance, *Junior age* credentials contain the attributes like ≥ 12, ≥ 18 etc., whose values will be *yes/no*. The age credential can be issued to the IRMA cardholder by an authorized issuer such as the municipality.

Junior Age credential	Senior Age credential
≥ 12	≥ 50
≥ 16	≥ 65
≥ 18	≥ 75
≥ 21	≥ 80

Table 3.1: Age credentials in IRMA

When the relying party or the verifier requires the cardholder to prove that she is over 18 years of age, he requests the card to reveal ≥ 18 attribute value from *Junior Age credential*. During the selective disclosure, the card generates a proof for the validity of the Junior age credential and reveals ≥ 18 attribute whose value will be either 'yes' or 'no'. The expiry date of this credential is also revealed along with the proof; so the issuer must take care while issuing not to encode the birth date of the cardholder as the expiry date of the credential as it will leak the birth date of the cardholder every time this age attribute is revealed thus harming the desired privacy. As an advancement to the existing approach, derived attributes can be used to prove the age (by deriving age from the *date of birth* attribute); however, then an additional proof has to be provided to prove that this new age attribute was indeed derived from the original *date of birth* attribute. Interval proofs could also be used in future to have more flexibility in proving age ranges but it will involve more complex operations on the card and these interval proofs are very computation-intensive.

Other data minimization functions like **Community ID/Residence postal code verification**, we can follow the below approach: IRMA card can have a *Address* credential and *Community ID* as one of its attributes. A selective disclosure can be done by the user revealing only the *Community ID* attribute from the *Address* credential. But as this reveals the *Community ID* value to the verifier we can add the attribute matching feature to IRMA card i.e. when the verifier requires the Community ID verification, he can send the particular value of the community ID to the card and the card does the matching. If the value sent by

the verifier matches the *Community ID* attribute value on the card, it reveals the Community ID attribute. Otherwise, the card hides the attribute while it still provides an empty proof to the verifier that credential is valid. This approach can be scaled to any number of attribute value verifications in the IRMA card. It preserves the cardholder's privacy as it does not reveal extra information and at the same time guarantees the validity of the credential and the revealed value.

3.6 Pseudonym generation in IRMA

In this section, we discuss the pseudonym generation in IRMA and how different is it from the pseudonym generation in nPA. As in nPA, IRMA also supports the pseudonym feature where the IRMA cardholders can choose to be identified to a service provider under a pseudonym. The service providers must provide this option with their digital service and they can also request the user to identify herself with a pseudonym when it requests for the required set of user attributes. But the service provider access rights policy must include the rights to access pseudonyms of the users.
The pseudonym and revocation operations in IRMA are performed in a group in which the DL (Discrete Logarithm) problem is hard.
A pseudonym is generated using two input values:
(1) a_{nym}: The personal pseudonym value for the owner of the card.
(2) A service provider-specific value; For instance, it could be the URI (Uniform Resource Identifier) of the web service of the service provider (unique for a service provider).

Here a_{nym} is like an attribute issued by scheme authority/pseudonym authority. The personal pseudonym value is signed by the issuer and stored on the card and it is specific to the user. It is different from a master secret or a static secret key stored on the chip of the card which is specific to the card. In IRMA, the pseudonym generation [40] is done in the steps listed below:

1. The card transforms the service provider's unique identifier suchas the URI received from the service provider into a generator g_{nym} which is an element of the DL group for pseudonym operations.

2. Next, the personal pseudonym value a_{nym} that is stored on the IRMA card is transformed into a service provider specific pseudonym ID_{nym} as shown in the equation: $ID_{nym} = {g_{nym}}^{a_{nym}} \bmod p_{nym}$.

An advantage of the IRMA pseudonym generation scheme is that the personal pseudonym value a_{nym} is an attribute just like other signed attributes on the card

and it is user-bound. The pseudonym ID_{nym} resulting from the above generation scheme does not make use of the master secret stored on the IRMA card chip, instead, it uses the unrevealed attribute a_{nym}. The pseudonym authority can maintain a record of pseudonym values corresponding to the issued IRMA cards' serial numbers. So, in the case of loss of an IRMA card, a_{nym} can be copied like any other attribute onto another blank IRMA card. Thus any number of identical pseudonyms can be provided for a user either if the card is lost or to use same pseudonym on various authentication devices.

3.7 Revocation of credentials in IRMA

The revocation scheme for IRMA that is currently being proposed by the IRMA team involves a semi-trusted party in the system, a *Revocation Authority (RA)* that is responsible for revoking the credentials. RA keeps track of the revocation values of revoked credentials. In this approach, the time is split into epochs and the RA chooses per-epoch per-verifier generators [41]. The setting is a cyclic group G with prime order q and every credential encodes a randomly chosen *revocation value* $r \in Z_q$. If a credential has to be revoked then the value r is added to the global revocation list RL. When the user shows the credential to a verifier, the verifier checks whether the user's revocation value r appears on the revocation list RL. The revocation scheme can be summarized as follows:

1. The RA sends a precomputed and a pre-sorted revocation list $RL_\epsilon = g_\epsilon^{r_1}, .., g_\epsilon^{r_k}$.
2. The verifier sends a per-epoch per-verifier generator g_ϵ to the user.
3. The user uses g_ϵ to create the verifier specific revocation token by embedding the revocation value r into a Revocation token R like this:
   $R = g_\epsilon^r$
4. The user then sends the revocation token R to the verifier.
5. Now the verifier just has to check if $R \in RL_\epsilon$.

Two of the restrictions presented on the scheme by Alpar et al. [41] are:

- The credential must be able to encode a revocation value r from a sufficiently large set. This value can identify a credential if it is revoked. The notation C(r) is used to denote a credential that contains the revocation value r. Depending on the type of credential (whether root credential or regular credentials), there may be other attributes present.

- The showing protocol must be extendible to provide the verifier with the revocation token R = $g_\epsilon{}^r$ and a proof that R and C(r) contain the same revocation value r. This can be done by using Zero-knowledge proofs on which IRMA relies on and can be extended readily to include the required proof of equality.

Advantages of the scheme:

- The use of card-specific revocation values and verifier-specific generators ensures privacy and cross-domain unlinkability.
- This scheme is applicable for multi-show unlinkable credentials as it uses per-epoch verifier generator and the verifier no more receives bare revocation values based on which he can track a user or link several transactions of the user made in different epochs i.e even if the user shows a credential to the same verifier multiple times during different epochs, then the verifier will not be able to link these showings and track the user via the credential. However, if the user shows the same credential more than once during the same epoch (session) then the credential showings can be linked.

Chapter 4

IRMA-based eID authentication

In this thesis, we propose an IRMA-based eID authentication approach which can serve as a replacement to the eID authentication function of nPA. We retain the sequence of the nPA authentication steps but the EAC protocols in nPA are modified in our approach according to IRMA's concepts like selective disclosure accompanied with the zero-knowledge proofs. The PKI infrastructure of nPA can be used for issuing terminal certificates to the relying parties; CVCA (Root CA) signs document verifiers (DV) or Relying party authorizers and each DV issues Terminal Authorization tickets to registered relying parties. The root CA's public key can be stored on the IRMA card by the Scheme Authority during its initiation. Figure 4.1 illustrates the communication flow between the card and the verifier during an online authentication session and the manner in which nPA's Terminal authentication and Chip Authentication protocols are replaced by IRMA protocols. We refer to the relying parties as *verifiers* from now onward and assume that the eID-Server that does the actual authentication by verifying the attributes and proofs sent by the IRMA card is running in the verifier's domain. It is an attached eID-Server and not run by any third party. So we refer to the verifier as the data requesting, processing and verifying entity. The authentication procedure with IRMA eID card is described in the following steps:

1. The user accesses an eService's website and decides to authenticate via her IRMA card. The eService transfers the control to the eID-Server which takes up the job of authenticating the user's attributes required for the current transaction by securely connecting to the IRMA-Client on the user side.

2. The verifier's authorization certificate is presented to the user at the local terminal. This certificate contains the verifier's public key, certificate, attribute access policy and eService's URI (Uniform Resource Identifier) that is used for pseudonym generation. By design, the IRMA card implementation allows only the verifier-requested attributes to leave the card based on

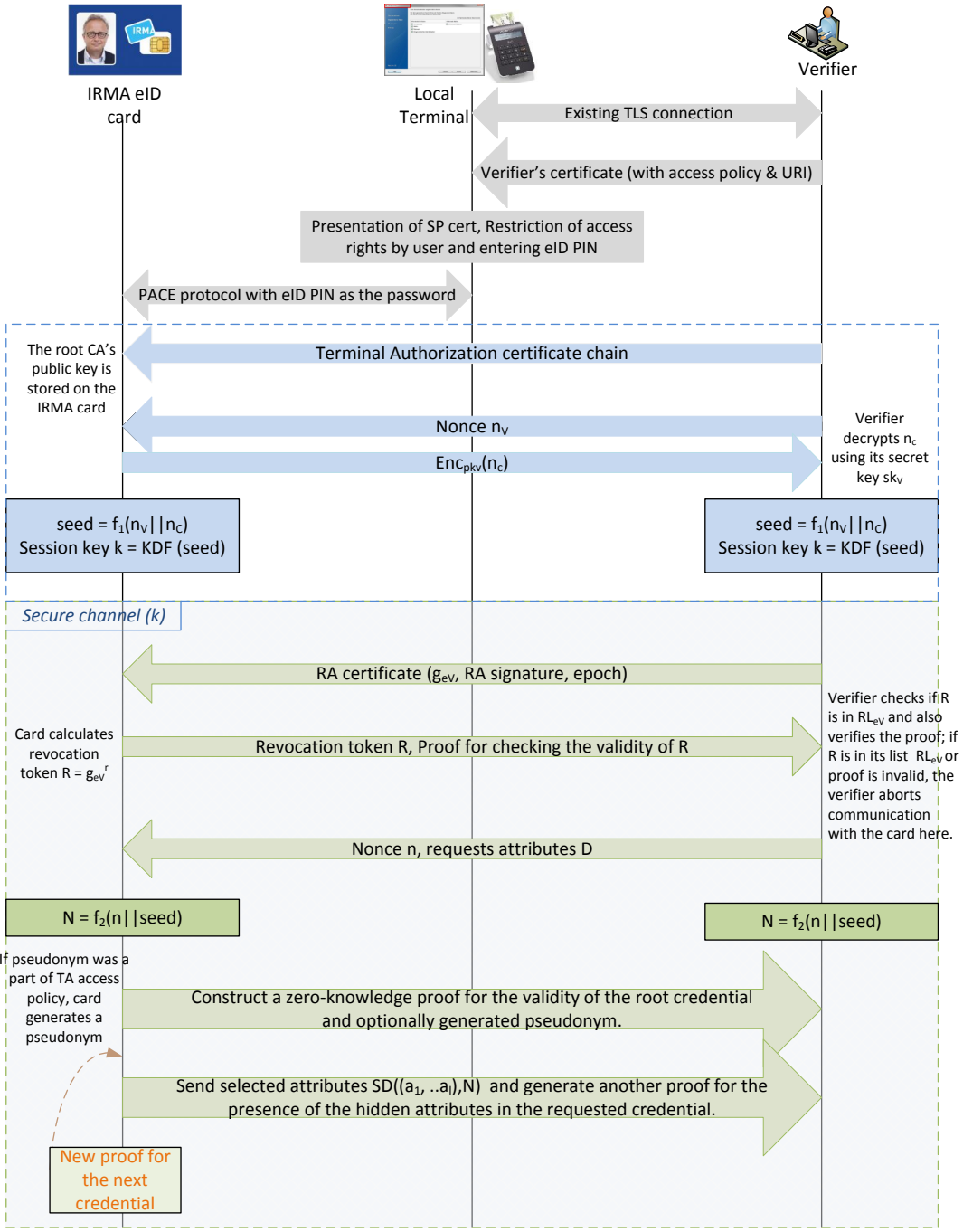

Figure 4.1: Flow diagram depicting IRMA Terminal Authentication, Chip Authentication and Selective attribute disclosure

the access policy stated in the terminal authorization certificate. A second level of user control enforcement could be added in IRMA similar to how

it is done in nPA i.e. by displaying the requested attributes to the user on the PC/tablet/phone screen by the IRMA client application and allowing the user to restrict the number of attributes further if she wishes, before entering the PIN code which initiates the PACE protocol.

3. In the case of an *offline authentication*, the user is authenticated to the verifier's terminal (IRMA card reader on the relying party side) by the initiation of the PACE protocol as the user presents her IRMA card with her photo on it and enters her PIN code. A direct end to end secure communication channel is established between the IRMA card and verifier with the aid of session keys generated during the PACE protocol. The PACE session keys secure the communication until the next set of session keys are established. In the case of an *online authentication*, PACE is used for PIN-sharing and to create a secure channel between the IRMA card and the local terminal (comprising of both the card reader and the IRMA-Client application) on the user's side. One of the preconditions is that there exists a TLS-secured channel between the local terminal and the verifier which is established by the IRMA client application. The local terminal is responsible for encrypting and checking integrity of the messages that are exchanged between the IRMA card and the verifier.

4. During the *Terminal Authentication (TA)*, the verifier terminal has to authenticate to the card, that it is an authentic certified terminal and has a valid authorization certificate (TA certificate). So the verifier sends the complete terminal authorization certificate chain (containing the root CA's public key) to the card. The IRMA card can verify the legitimacy of the verifier's TA certificate and its authorization with the help of root CA certificate stored on its chip. This step verifies that the public key of the verifier is authentic and it is issued by the authorized party (CVCA).

5. In order to make sure that the verifier is indeed who it claims to be, the card has to verify if the verifier has the private key corresponding to the public key. In IRMA, this is done by *nonce generation*. The verifier generates a nonce n_V and sends it to the card. The card encrypts its nonce n_C with the verifier's public key pk_V that is sent as a part of the TA certificate. The verifier now needs to decrypt using its private key to obtain n_C. Once both the parties have n_C and n_V, a seed is computed by concatenating these nonces and on applying a key agreement function on this seed, both the card and the verifier will agree on a session key k which will be used to encrypt the further communication between them.

6. *Chip Authentication and Passive Authentication protocols used in nPA can*

be collectively carried out by the selective disclosure step in IRMA.
This is done within the secure channel established by the session key k agreed by both the card and the verifier in the previous step. Coming to the chip authentication phase, the IRMA card needs to authenticate to the verifier if it contains an authentic chip which is officially issued by the Issuing Authority (could be the Government). In the case of IRMA, privacy and anonymity are not achieved here by sharing the key with a group of IRMA cards as it is done in nPA. Instead, there is only one user-bound master secret s stored on the IRMA card. Each IRMA card can prove the presence of its master secret with the help of a *zero knowledge proof*; the proof is generated during the selective disclosure step and sent to the verifier.
We can have a root credential on an IRMA card that consists of this master secret s, pseudonym value a_{nym}, revocation value a_{rev} and credential expiry date a_{exp} as attributes and this root credential is signed by the Scheme Authority (or an authorized credential issuer). Other credentials contain the issuer's signature to affirm that the card chip indeed holds this master secret s along with their respective expiry dates and attributes. In IRMA, valid credentials and the issuer's signature vouch for the authenticity of the chip as well as the integrity of the attributes stored within the credentials. It is known as **Implicit Chip Authentication (ICA)**. The selective disclosure process in IRMA replaces the passive authentication procedure of nPA as the presence of a valid Issuer signature on the credential and a valid proof for the same credential implies that the credential has been issued by the authorized issuer and the attributes within that credential not been changed or tampered in any way since its issuance. So this process successfully checks the integrity of the attributes that are exchanged with the relying party. The two main steps carried out within the session key k-secured channel are:

- *Revocation check:* The verifier sends its Revocation certificate (RA certificate) issued by the Revocation Authority to the card. It contains the verifier-specific epoch-specific generator $g_{\epsilon V}$, epoch value and signature of RA. On receiving this value, the IRMA card calculates card-specific revocation token R by combining its own revocation value and the verifier's generator $g_{\epsilon V}$. This revocation token R and a proof is sent to the verifier to prove that the card has a valid revocation value that was used to create the token R. The verifier checks if this R matches any entry in its Revocation List, it has to just check if $R \in RL_{\epsilon V}$. If a match is found, then the verifier concludes that the card has been revoked, rejects all other received values from that IRMA card and aborts the authentication process. Otherwise, it proceeds to the selective disclosure step. The validity of the revocation token R generated by the card

during this step can be verified by the verifier with the help of the proof generated by the card and sent along with the revocation token.

- *Selective disclosure of the attributes and generation of zero-knowledge proofs*: The verifier sends a nonce n and attributes to be disclosed D. At this point, both the card and the verifier calculate a new nonce N as $N = f(n \parallel seed)$. The purpose of the nonce N is to check the freshness of the card-generated proof and to bind multiple proofs to the same session. During the selective disclosure,
  - If pseudonym was a requirement stated in the verifier's access policy that was initially sent to the card, the IRMA card generates the pseudonym by combining its own personal pseudonym value (a_{nym}) and the verifier-specific value (e.g. URI of the verifier). Along with this, a proof is generated for the validity of the root credential that contains the pseudonym value a_{nym}, revocation value a_{rev}, secret s and credential expiry date a_{exp}. This proof should be verified by the verifier, if it is valid then the verifier is convinced of the presence and validity of the pseudonym and also the revocation token R that was generated by the card during the revocation check. Thus, if pseudonym generation is done then the proofs for both pseudonym and revocation values can be combined as both a_{nym} and a_{rev} values are stored within the root credential.
  - Then, the card sends the disclosed attributes along with expiry date of the credential and generates another zero-knowledge proof for proving the validity of the credential containing the attributes and to prove that the disclosed attributes and the hidden attributes are indeed a part of this credential.
  - The verifier proceeds with verifying the received attributes and validity of the proof. If the proof is valid, the verifier accepts the disclosed attributes and authenticates the IRMA card successfully. Otherwise, the verifier aborts the authentication process.

4.1 IRMA eID system infrastructure

The IRMA-based eID system needs an infrastructure around it and we have come up with an infrastructure as shown in Figure 4.2. It consists of a Scheme Authority (SA) at the centre. It can be the government itself or a government-operated entity that carries out all the responsibilities mentioned in the Section 3.1.1. Authorized ID manufacturers manufacture the cards in which the master secret key is generated for each card and send these cards to the SA. Once the secret keys are

generated on the cards, they are not transmitted to any entity within or outside the eID system and these secret keys may never leave the card. There are two semi-trusted parties namely Pseudonym Authority (PA) and Revocation Authority (RA) who provide personal pseudonym and revocation values respectively for the IRMA eID cards. The SA is responsible for loading the pseudonym and revocation values along with its own public key (root CA public key) onto the IRMA cards before issuing them to the users. It is also in the hands of SA to choose different Relying party authorizers and Credential Issuer authorizers.

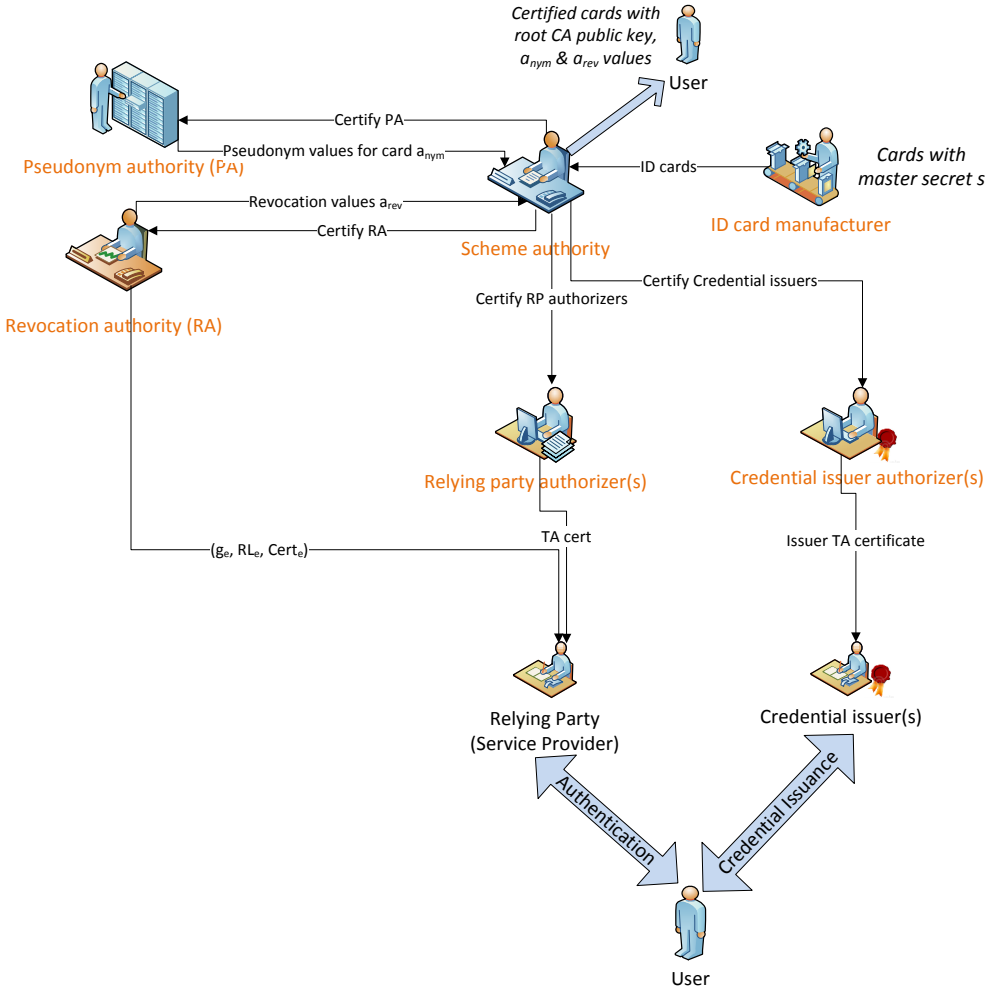

Figure 4.2: Schematic diagram of the proposed IRMA-based eID authentication infrastructure

4.2 Security analysis and advantages of the IRMA-based eID authentication

In the IRMA eID authentication approach, the eID data on the card are stored as attributes inside a cryptographic container called credentials. Every credential on the card carries a certified credential issuer's signature. During the authentication phase, the security features provided by such an approach are as follows:

- As we retain the PACE protocol from nPA, authentication of the user to the local card reader as being legitimate card owner is taken care of and the card PIN or the data is not sent unencrypted over air so, the data is protected from sniffing and eavesdropping.
- The verifier's (service provider) terminal authentication is done by a nonce exchange between the card and the verifier and eventually agreeing on a common session key. The *confidentiality and tamper-resistance* of the verifier's public key and access policy within the sent terminal certificate is preserved by the TLS-secured channel. In case of an offline scenario, the communication between card and the verifier's terminal is protected by the secure channel setup by the PACE session keys.
- The card cross-checks the access rights of the verifier's terminal mentioned in the TA certificate with the requested set of attributes. Accordingly, a non-interactive zero-knowledge proof can be produced by the credential holder who reveals the desired subset of the attributes along with the issuer's signature and a proof that the disclosed attributes are actually present in the credential.
- In IRMA, valid credentials and the issuer's signature vouch for the authenticity of the chip as well as the integrity of the attributes stored within the credentials. Brickel et al. give an explanation for this method's authenticity in their paper [42] and the points that hold even for IRMA are:
  - As IRMA uses Camenisch-Lysyanskaya (CL) signature scheme, respective discrete logarithms based proofs prove the possession of valid root credential.
  - Unforgeability of the credentials hold under the strong RSA assumption.
  - Privacy and anonymity are guaranteed under the Decisional Diffie Hellman (DDH) assumption[1].

[1] A definition for DDH assumption as stated in Wikipedia is: The decisional Diffie–Hellman (DDH) assumption is a computational hardness assumption about a certain problem involving discrete logarithms in cyclic groups.

- The *confidentiality* of the revealed attributes are preserved as they are sent through the secure channel established by the session key 'k' (See Figure 4.1) and the notion of *indistinguishability* between the session key and a random string of the same length is captured in this protocol.

- Every attribute that is revealed carries its issuer's signature making it easy to verify its integrity. Also, the verifier who receives the signed attributes from the card cannot present those attributes as valid attributes to another verifier posing as the real owner of the attributes because he will not be able to generate the zero-knowledge proof for those attributes.

- The verification instances carried out by the relying party using a credential are unlinkable, unless the revealed attributes make them linkable. A verification instance is also unlinkable to the issuance of the credential (for example, when the credential issuer itself becomes the verifier).

- An adversary cannot forge an authentication by using a saved matching conversation from another session within the secure channel established by the session key k because the value $N = f_2(nonce\ n \parallel seed)$ not only ensures the freshness of the proof generated by the card but also binds the selective disclosure proof to the ongoing secure session. Moreover, N cannot be existentially forged without knowing the *nonce* n and the *seed*.

- The IRMA pseudonyms provide cross-domain unlinkability as it uses card-specific service-provider specific pseudonyms.

- Epoch-based verifier-specific revocation tokens enable cross-domain unlinkability and also multi-show unlinkability when credential is shown to the verifier at different epoch times.

- The history function is implemented in IRMA's card management app in order to ensure better accountability of transactions for the users.

4.2.1 Advantages of IRMA-based eID authentication approach

- **No more group keys:** The group key (as mentioned in the Limitations of nPA in Section 2.8.1) used for millions of citizens in nPA can be replaced by a *user bound key* in IRMA. The notion of signed valid credentials and dedicated secret key for each eID card will provide more security and assurance to the eID systems and privacy to the citizens. Our approach uses Implicit Card Authentication (ICA) for chip authentication purpose; A card can also

generate an empty proof i.e. without revealing any attributes to prove just the validity of the card. This facilitates easier blacklisting of the eID cards without having to blacklist a batch of cards which share the same private key.

- Binding of items like the user's identity, pseudonyms to the eID card is protected only by the chip in the card which is the secure element in the case of nPA; the algorithms/functions used for the age or the community ID verification run on the chip and the output of this function is blindly trusted as it is being transmitted through the secure channel. None of the individual attributes are signed by the issuer in nPA; this signature would have enabled direct verification of the authenticity of the data transmitted from the eID card. In IRMA all the credentials in the card are bound by the signature over the attributes. If an adversary comes in possession of the card and tries to harm the integrity of the information present on the chip by directly tampering with the chip, it would invalidate the signature. This provides an added security to all the data stored inside an eID card.

- This approach achieves maximal privacy as the real identity of the card owner is never released or transmitted to the relying party during an authentication session unless the revealed attributes are uniquely identifying attributes. When the pseudonyms are to be used, the transformed non-real identity of the card owner is transmitted to the relying party.

- IRMA discloses just the necessary attributes while proving the presence of both revealed and hidden attributes in a credential and the randomized signature by carrying out a zero-knowledge protocol. Thus, it provides authentication of the card and credentials to the relying party while preserving privacy and anonymity. Multi-show unlinkability is achieved in all the verification instances and transmission of revocation tokens (within different epochs). Issuer unlinkability is provided by IRMA as it uses blind signature protocol during credential issuance.

- IRMA provides better flexibility than in nPA in the following ways:

  - Personalized and individual attribute authentication enables better user-control.

  - Several authorized credentials issuers can issue credentials on the same IRMA eID card that reduces number of cards that a user has to carry for various services. In this way it provides flexibility in the use of eID in both public and private sectors.

– IRMA technology could be used with any Android-based terminal. The usage is not limited to PC. So IRMA cards are suitable even for many offline use-case scenarios where the terminals just need to be preconfigured with its own terminal certificate and they must have stored the set of issuers' public keys. Then those terminals can verify the IRMA cards without being connected to the Internet.

– Another main advantage of IRMA pseudonyms is that the IRMA card users can obtain identical pseudonyms over multiple authentication terminals for a registered online service or during the situations when a new eID card is issued because the previous one was revoked due to loss or expiration.

4.2.2 Drawbacks of IRMA

IRMA is a new technology that has been developed and constantly being updated by the IRMA team at the Radboud University Nijmegen. Although it is an attractive option to be merged into an eID system for privacy-preserving authentication, it has some drawbacks which are listed below:

- IRMA is not a very mature technology, it lacks experience and usage in the past that would serve as evidence for its success. Although attribute based credentials have been discussed in theory for more than a decade now, a technology like IRMA that employ ABCs on smartcards with a reasonably good performance for practical use is just one year old [5]; the system around it is not yet well developed to support a national level eID system.

- The number of resources that IRMA requires can be a limitation. If a larger number of attributes has to be stored in a single credential, more memory is required which is not ideal for a smartcard implementation. At the moment, IRMA allows only for a maximum of 5 attributes within a single credential, each one up to maximum 255 bytes in size because of the available RAM (less than 2kB) on the card. The device or the smartcard will run out of memory if more attributes are stored. Moreover, the time needed for the attribute disclosure also increases if many attributes have to be hidden while revealing the selected attributes. The transaction times get added up if many attributes from different credentials have to be revealed at once in the case of existing dependencies. As the ABC proofs are rather expensive in terms of time, it is often desirable to omit as many selective disclosure proofs as possible in practice [43].

- The prime modulus size used in IRMA is 1024 bits. This modulus size is not sufficiently large to ensure higher security but this is also an issue raising out

of smartcard resource limitations. The currently available smartcards do not have sufficiently large memory to store all the intermediate values required for the proof generation or the power to handle all the complex modular computations in a reasonably low time. This, however, is not a fundamental limitation. If there is a market for such powerful chips with higher memory, they will be produced.

4.3 Performance considerations

4.3.1 The German eID performance

In order to assess the performance of the German eID card's online authentication, we carried out an instance of eID authentication using the eID card, basic reader and the Ausweisapp. Then we noted down the time taken for one authentication session which came upto 5 seconds. We also collected the packet trace of the USB communication between the card and the reader using Wireshark and tried to analyse the time taken from the time stamps for the corresponding card-reader command and response. From this, we could deduce that the entire session took 5 seconds to complete and this calculated time matched with the time observed during a practical authentication session.
The PACE protocol itself takes 1 second and this was cross checked with the performance measures given in the PACE document [16] by Bender et al. With the help of timestamps provided in the packet trace, we calculated the nPA total execution time by summing up the times taken both by the card and the host terminal. It came upto **5.65 seconds** out of which the time taken just for the card operations and response generation is **3.12 seconds** .

4.3.2 IRMA - Idemix performance

In this section, we refer to the most recent work of Pim Vullers [33] for the performance of smart-card implementation of attribute-based credentials. There are two important performance measures: the time it takes to issue a new credential to the card and the running time of the verification protocol. The curent Idemix implementation of IRMA uses a modulus size of 1024 bits, which provides a minimal level of security, but an acceptable performance. Given this size of the modulus, and hence the size of all group elements, at most 5 attributes can be accommodated per credential in this implementation. From the performance results stated in the Idemix chapter, Performance results section of Pim Vuller's PhD thesis [33], the following deductions about the execution times can be made and the times considered here are the sum of computation times and the involved overhead.

1. **Idemix credential Issuance times:**
   Time taken to issue

   - 5 attributes is 2.60 seconds
   - 4 attributes is 2.52 seconds
   - 3 attributes is 2.45 seconds
   - 2 attributes is 2.37 seconds
   - 1 attribute is 2.26 seconds

   Deduction: The number of attributes included in a credential has only a small effect on the computations. An increase in the number of attributes does, however, result in an increase of the overhead of approximately 100 milliseconds per attribute.

2. **Idemix credential verification times:**
   Time taken to selectively disclose # of attributes in a configuration where there are 4 stored attributes in a credential are:

   - 1 attributes is 1.20 seconds
   - 2 attributes is 1.13 seconds
   - 3 attributes is 1.01 seconds
   - 4 attributes is 0.93 seconds

   Deduction: Each attribute that is disclosed during the selective disclosure reduces the computation time with roughly 100 milliseconds. The time taken increases with the increase in the number of attributes to be hidden.

Calculation of total execution time for a IRMA-based authentication session: The total execution time taken for authentication in IRMA is the sum of execution times for PACE, IRMA's Terminal authentication, Implicit card authentication, revocation check and selective disclosure function (comprising of attribute disclosure and zero-knowledge proof generation). The time is calculated based on the information given about RSA and AES execution times on NXP Contactless cards given in the NXP Document 75016728[2].

1. PACE protocol execution: In the case of IRMA, Elliptic curve cryptographic operations are not used instead, we use the normal Diffie Hellman key agreement so we expect the time taken by the PACE protocol in IRMA will be less than 1 second.

[2] http://www.nxp.com/documents/line_card/75016728.pdf

2. TA: With minimum two exponentiation operations, AES-128 encryption and decryption operations and one hashing function SHA-256 execution. For one RSA-1024 bit operation, the NXP contactless card chip takes 0.071 s, AES-128 and SHA-256 operations takes 0.000012 s each; so total time for TA = 0.142 s + 0.000036 s = 0.142 seconds

3. CA: Time for the revocation check + Selective disclosure = 0.142 s + 1.2 s = 1.342 seconds

We expect this time to be approximately 2.5 seconds and an additional 2 seconds for non-cryptographic functions that use a slower CPU on the card. In total the IRMA authentication time will be **4.5 seconds**. So, performance-wise IRMA is not too far behind nPA.

4.4 Use cases for the proposed system

The privacy protection offered by using attribute based credentials (ABCs) is significant for several use cases. In this section, we describe some use-case scenarios where IRMA-based authentication might prove very appropriate as it provides better privacy and flexibility when compared to the federal German eID card authentication.

4.4.1 Age proof scenarios - Offline or online

The most common and prototypical application of a privacy friendly authentication mechanism is to prove the age without revealing any other information. To buy cigarettes or alcohol in the Netherlands, one needs to prove that she is at least 16 years old. To buy strong liquor, you need to prove you are at least 18 years old and to get reduced fares in public transport you need to prove you are at least 65 years old. In the above cases, "at least 16", "at least 18" and "at least 65" can be encoded as respective attributes on the IRMA card whose disclosure will prove that the cardholder satisfies the required age criteria. Another important aspect in this sort of age verification is that the absence of an "at least 16" attribute is no proof of being at most 15, as people may choose not to disclose that attribute. So only a positive attribute can prove you are below 16; for instance, in a scenario where the cardholder wants to join an online forum for children below 16 years of age. The fact that an IRMA card carries the picture of the holder allows the use of the IRMA card for such use cases offline as well.

4.4.2 Use cases based on eligibility criteria other than age

Some of the use case scenarios that involve meeting certain eligibility criteria can fully benefit from the the privacy features of the IRMA-based authentication approach just like the age verification. They are briefly mentioned below:

- **Corporate scenario: Employee credential check**
  An employee of an organization can just prove yes/no to the "is an employee" and if required "privilege" attribute to satisfy an organizational eligibility criteria in order to enter a building or access an online resource.

- **Healthcare sector: Checking the doctor's eligibility to prescribe certain medication online**
  If a doctor is prescribing medication/drugs of a specified dose or quantity to his patients (online), the doctor can use his IRMA card to authenticate to the medical association or even the pharmacy that he is eligible to do so. In this case, he can just prove his eligibility by revealing his "academic degree" attribute. However, we do not expect anonymity in such a use case as identification and traceability of the doctor who prescribed certain medicine to his patients becomes a requirement. For this purpose, we can make some identifying attributes 'mandatory' such as the doctor's name and his registration ID. But still the doctor can wish to hide other attributes like his "speciality" attribute or his "membership" to some medical authority/standardized body; such attributes can be hidden if with the selective disclosure property of IRMA.

4.4.3 Service Subscriptions

In the case of subscriptions to an online service, authentication is required and the IRMA based approach can be used to carry out this kind of authentications as it provides security and preserves the privacy of the subscriber. Here, an attribute can encode the access rights that subscriber holds, like 'Active subscription'. Taking the example of online newspaper subscription, authentication can be done via the IRMA card without revealing any information other than the attribute 'Active subscription' whose value will be 'yes/no'. This not only minimizes the data that the subscriber has to share with the online news service provider but also prevents the user's login/ read patterns to be complied. The user can also use the service-specific *pseudonyms* while gaining access to their respective subscriptions; in the case of IRMA, the user can still retain her pseudonym if she loses the card and access all her subscribed services once she has the new IRMA card. In this way, user will not lose her record of previous transactions made with a service provider nor will she lose money if she already had a yearly or a bi-yearly subscription

under a pseudonym. There can be several possible use-cases similar to the online news subscription example and IRMA card also provides the flexibility for the authorized service providers to issue their own credentials to their customers.

4.4.4 Purchase of tickets for an event

In the online ticket sales scenario, the following issues can be seen:

- Existing offline and online ticket sale mechanisms do not encourage preserving the privacy of the buyers. The personal data collected during the buying and showing the tickets at the event reveals more-than-required information about the buyer.
- There is also a risk of illegal resale of the event tickets which is apparently very rampant in the Netherlands. Professional resellers buy the tickets in bulk as soon as they are for sale. They do so either offline or online. Once the concert tickets sell out, the resellers will sell their tickets at a huge profit. It is hard to stop this fraud even when the sale of online tickets is bound to personal accounts, with a limit of say 4 tickets per account as the resellers have found ways to overcome such barriers.

With the IRMA approach, a concert ticket can be considered as a credential with several attributes, including the date and title of the event for example, the number of people the ticket is valid for and possibly a sequence number. To buy a ticket online, the IRMA cardholder inserts her IRMA card in a smart card reader attached to her PC or puts her IRMA card against the back of her NFC-enabled phone (IRMA cards are contactless). The online ticket office sells the tickets online as credentials that are issued to your IRMA card. In fact, the ticket office web server connects to your card to upload the credential once the transaction is approved. In the offline case, you insert your card in a terminal at the ticket office, and the process is pretty much the same after that.

The ticket that is uploaded to the smart card as an attribute is later shown at the entrance gate of the event. The doormen will compare the user with the picture on her card and then verify whether she has the right credential for the event on her IRMA card. They could do so with a special application installed on their NFC phone.

Benefits of using IRMA in this use-case scenario are:

- Minimum information is collected from the buyer. If the payment method does not reveal the identity of the buyer, the online seller cannot compile a profile of all the tickets a particular person bought.

- The access to credentials is restricted, so the doormen can only access the credential for this particular event and no other credentials on the IRMA card. So, the confidentiality of the data on the data and the privacy of the buyer is preserved.
- The tickets cannot be resold because credentials are bound to a private key stored securely in the card and IRMA cards are personal.

Chapter 5

Conclusion & Recommendations

5.1 Conclusion

Internet is a shared global resource that needs to be protected for the good of people and the society including the critical infrastructure and individual privacy. National identification management systems have set their transition from paper-based to electronic ID cards for the citizens and enabled them for identifying the citizens on the Internet. While carrying out online e-government and private business transactions, security, privacy and scalability play a very crucial role for all the involved parties.

In this thesis, we discussed in detail the most recent and advanced eID system deployed in Europe which is the German eID system (nPA); we analysed its eID functionality, cryptographic protocols used, the security and privacy design features and their limitations. From this analysis, we found out that there are serious disadvantages arising from some of the nPA's design features like *the shared key among a large batch of eID cards for preserving privacy*, *complete trust on the security and tamper-resistance of the card's chip and the secure channel set up by the chip* and *no authorized signature on the eID data that is being transmitted in the trusted channel*. The risk of totally depending on the chip's secure element and the shared key for millions of cards is that if a single eID card's chip is compromised then, all the chips sharing the same key should be revoked; this might cause a major disruption in the eID system operations, even resulting in loss of trust in the entire eID system. Therefore, a different design methodology is needed for the security and privacy features in nPA; an incremental improvement in nPA will not be sufficient. Furthermore, there is a need for some improvement in terms of flexibility and usability of an eID for the citizens. The nPA being a federal ID lacks flexibility when it comes to certain aspects like the impossibility to create identical

pseudonyms for users in the case of lost, expired or stolen cards and limited possibilities for the private service providers to issue their credentials to the citizens on this eID card. It is important to consider both eGovernment and eBusiness data transactions with the eID card from an economical perspective. In the case of national eID systems, it is also preferred to integrate the extended privacy goals like multi-show unlinkability and transparency to the traditional security-preserving goals and generate awareness among the citizens about the privacy issues while balancing the interests of all the parties involved in the eID system.

Keeping all the above points in view, in this thesis, we explained the concept of Attribute-based Credentials (ABC) and proposed the IRMA technology which is built on the principles of the ABC systems as a plausible technology to be integrated into an eID system such as nPA for the purpose of eID authentication. We analysed the features provided by IRMA and how feasible the eID authentication with IRMA will be and if it will overcome the prevailing disadvantages of nPA. Table 5.1 gives an succinct comparison of the features supported by nPA and IRMA.

	nPA	IRMA (Idemix)
User control-selective disclosure	✓	✓
Data minimization functions	✓	✓
Signed eID data/attributes		✓
Keys used	Shared key for a large batch of cards	User-bound secret key per card
Pseudonyms	✓	✓
Identical pseudonyms (if the card is lost)		✓
Cross-domain unlinkability	✓	✓
Multi-show unlinkability		✓(limited[1])
Digital signature (QES) function	✓	Not supported at the moment.
Implementation on the smartcards	✓	✓
Readiness of deployment	✓	

Table 5.1: Comparison between the nPA and IRMA features

[1]Multiple showings of a single credential is unlinkable between epochs, however, they are linkable within the same epoch.

From the above table, we can conclude that in comparison with nPA, IRMA provides the following advantages:

- Increased privacy by the use of *zero-knowledge proofs and user-bound keys*. IRMA also provides *multi-show unlinkability* in addition to the single-show unlinkability (cross-domain unlinkability as provided by nPA).
- Better flexibility to an eID system in terms of *attribute issuance*, *selective attribute disclosure* and provision for generating *identical pseudonyms* if the card is lost, expired or stolen.
- Increased security *by not tying the entire security to the secure element on the chip* and having *signature over every attribute* bound to the card.
- Ability to revoke just a single card as no group key is used in IRMA.

In conclusion, a potential future deployment of IRMA-based eID authentication in the eID systems would allow them to go beyond the flexibility, security and privacy-preserving capabilities of the German eID model. However, there are some limitations of IRMA at present; the main one being *IRMA's novelty and the absence of an established infrastructure or a complete proven system around IRMA*; this could be a reason why the government may not be able to move ahead with merging IRMA into a national eID system. But once a system is designed and developed with IRMA technology and it is deployed in some other medium scale projects, for e.g. City pass, IRMA will gain some sort of credibility and maturity. Then, it can be considered as a more suitable option for the national level eID cards.

5.2 Future recommendations

Other than extending the IRMA pilot project to other medium scale projects and eventually realizing the final aim of deploying IRMA in the eID systems, future recommendations for an enhanced eID scheme proposed in this thesis are summarized below:

- In both nPA and IRMA-based eID systems, we have an eID card that should be used with an external reader device. While it is better to use an advanced reader device with its own keypad, we recommend an eID card to be a smartcard with an built-in keypad or gesture pad that eliminates the need for an expensive card reader with a keypad or an external terminal required for entering the PIN code.

Figure 5.1: New NXP smartcard with in-build keypad

Figure 5.1 is an NXP smartcard prototype[2]. From a security perspective, the type of smartcard suggested in this recommendation prevents the risk of an adversary obtaining the PIN code (or the eID password) entered by the user by installing a keylogger on the external reader or the PC connected to it. From a business perspective, it comes at a lower cost and allows the users to use inexpensive basic card readers, thus we can expect higher acceptance for these smartcards.

- In spite of designing such advanced eID schemes and measures to preserve privacy and eliminate traceability, in the online authentication scenarios, the users can still be traced based on the IP address of the connection being used. This risk could possibly be eliminated if the online authentication is performed over an anonymizing network, like the Tor[3] network. Further research in this direction may lead to finding a better solution to prevent tracing the users based on their device IP address.

- Although IRMA provides higher privacy benefits to the users than nPA, the security provided by IRMA can be increased by increasing the RSA modulus size to 2048 bits. However, cards with higher computing power and memory are required to realize this recommendation.

- The digital signature functionality is currently not supported by IRMA. A separate application should be used for creating digital signature in addition to the IRMA authentication. This poses a threat if a standalone naively

[2] http://www.engadget.com/2012/01/11/nxp-gesture-smart-card-nfc/
[3] http://en.wikipedia.org/wiki/Tor_(anonymity_network)

implemented signature application is integrated that uses a unique identifier for signing which violates the principles of user privacy and unlinkability that are the goals of IRMA. So we recommend for IRMA to develop and integrate digital signature functionality, that retains the the privacy-friendly structure of ABCs and provides the following features:

  - Ability to sign under a specific pseudonym without revealing any extra information.
  - Context separation i.e. the ability to sign under public and private contexts separately. For instance, a notary can use his signing key to sign both offical and his private documents by revealing only the corresponding attributes based on the type of transaction.

- IRMA could be implemented not only in the smartcards but on other platforms such as the SIM cards in the future and the support for using IRMA for authentication with different authentication instruments/tokens (for instance, mobile phones itself as authentication tokens) will increase the application scenarios for IRMA-based identification and authentication.

Bibliography

[1] BSI, "Innovations for an eID Architecture in Germany." http://www.personalausweisportal.de/SharedDocs/Downloads/EN/Flyers-and-Brochures/Broschuere_BSI_innovations_eID_architecture.pdf?blob=publicationFile, 2010. Online, accessed 26-February-2014.

[2] BSI, "Advanced Security Mechanisms for Machine Readable Travel Documents - Part 2," Tech. Rep. TR-03110-2, Bundesamt für Sicherheit in der Informationstechnik, 53133 Bonn, March 2012.

[3] I. Naumann and G. Hogben, "Privacy features of european eID card specifications," *Network Security*, vol. 2008, no. 8, pp. 9–13, 2008.

[4] J. Camenisch, I. Krontiris, A. Lehmann, G. Neven, C. Paquin, K. Rannenberg, and H. Zwingelberg, "Architecture for attribute-based credential technologies," *ABC4Trust deliverable D*, vol. 2, 2011.

[5] P. Vullers and G. Alpár, "Efficient selective disclosure on smart cards using idemix," in *Policies and Research in Identity Management*, pp. 53–67, Springer, 2013.

[6] W. Mostowski and P. Vullers, "Efficient u-prove implementation for anonymous credentials on smart cards," in *Security and Privacy in Communication Networks*, pp. 243–260, Springer, 2012.

[7] A. Menezes and P. Oorschot, "Handbook of Applied Cryptography," *Vanstone SA (1997)*, 1997.

[8] Modinis-IDM-Consortium *et al.*, "Modinis study on identity management in egovernment," *Common terminological framework for interoperable electronic identity management—Consultation Paper*, vol. 2, 2005.

[9] W. Fumy and M. Paeschke, *Handbook of eID Security*. Wiley, 2010.

[10] NXP Semiconductors, "The future of National eID: Increased security and citizen-centric services." http://www.nxp.com/documents/leaflet/939775017234_V9.pdf, 2012. Online; accessed 25-February-2014.

[11] M. Margraf, "The new German ID card," in *ISSE 2010 Securing Electronic Business Processes*, pp. 367–373, Springer, 2011.

[12] Government of the Netherlands, "Stakeholders, belangen en ontwerpeisen - Programma eID," tech. rep., Ministerie van Binnenlandse Zaken en Koninkrijksrelaties, Den Haag, January 2014.

[13] M. Harbach, S. Fahl, M. Rieger, and M. Smith, "On the acceptance of privacy-preserving authentication technology: The curious case of national identity cards," in *Privacy Enhancing Technologies*, pp. 245–264, Springer, 2013.

[14] T. Dierks, "The transport layer security (tls) protocol version 1.2," 2008.

[15] A. Poller, U. Waldmann, S. Vowé, and S. Türpe, "Electronic identity cards for user authentication—promise and practice," *IEEE Security & Privacy*, vol. 10, no. 1, pp. 46–54, 2012.

[16] J. Bender and D. Kügler, "Introducing the pace solution," *Keesing Journal of Documents & Identity*, vol. 30, pp. 26–29, 2009.

[17] BSI, "Architecture electronic Identity Card and electronic Resident Permit," Tech. Rep. TR-03127, Bundesamt für Sicherheit in der Informationstechnik, 53133 Bonn, March 2011.

[18] Ö. Dagdelen, "The Cryptographic Security of the German Electronic Identity Card," Master's thesis, Radboud University Nijmegen, The Netherlands, 2013.

[19] BSI, "Signaturgesetz vom 16. mai 2001 (bgbl. i s. 876), zuletzt geändert durch artikel 4 des gesetzes vom 26. februar 2007 (bgbl. i s. 179)," tech. rep., EU, 2007.

[20] J. Bender, D. Kügler, M. Margraf, and I. Naumann, "Privacy-friendly revocation management without unique chip identifiers for the German national ID card," *Computer Fraud & Security*, vol. 2010, no. 9, pp. 14–17, 2010.

[21] A. Pfitzmann and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management," 2010.

[22] C. J. Dietrich, C. Rossow, and N. Pohlmann, "Electronic ID card Online Authentication Network Threat Model, Attacks and Implications," in *19th DFN Workshop 2012*, 2012.

[23] D. Hühnlein, J. Schwenk, T. Wich, V. Mladenov, F. Feldmann, A. Mayer, J. Schmölz, B. Bruegger, and M. Horsch, "Options for integrating eid and saml," in *Proceedings of the 2013 ACM workshop on Digital identity management*, pp. 85–96, ACM, 2013.

[24] D. Hühnlein, D. Petrautzki, J. Schmölz, T. Wich, M. Horsch, T. Wieland, J. Eichholz, A. Wiesmaier, J. Braun, F. Feldmann, *et al.*, "On the design and implementation of the Open eCard App.," in *Sicherheit*, pp. 95–110, 2012.

[25] Kommune_Report, "Is the NPA better than its reputation?." http://www.kommune21.de/meldung_15186_Ist+der+nPA+besser+als+sein+Ruf%3F.html, 2013. [Online; accessed 20-January-2014].

[26] Keentech_Report, "eID acceptance in the NPA is lagging behind expectations." http://www.keentech.de/2011/01/eid-akzeptanz-im-npa-bleibt-hinter-erwartungen-zuruck/, 2011. Online; accessed 20-January-2014.

[27] H. Zwingelberg and M. Hansen, "Privacy Protection Goals and their implications for eID systems," in *Privacy and Identity Management for Life*, pp. 245–260, Springer, 2012.

[28] ABC4Trust, "Tutorial on Attribute-Based Credentials." http://www.dime-project.eu/en/Home/dime/events/list/tutorial-on-attributebased-credentials, 2011. Online; accessed 26-March-2014.

[29] S. A. Brands, *Rethinking public key infrastructures and digital certificates: building in privacy*. MIT Press, 2000.

[30] IBM Research Zürich Security Team, "Specification of the Identity Mixer cryptographic library," tech. rep., IBM Research, Zürich, 02 2012.

[31] L. Camenisch, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *Advances in Cryptology EUROCRYPT 2001*, pp. 93–118, Springer, 2001.

[32] J. Camenisch and A. Lysyanskaya, "A signature scheme with efficient protocols," in *Security in communication networks*, pp. 268–289, Springer, 2003.

[33] P. Vullers, *Efficient Implementations of Attribute-based Credentials on Smart Cards*. PhD thesis, Radboud University Nijmegen, The Netherlands, 2014.

[34] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[35] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology—CRYPTO'86*, pp. 186–194, Springer, 1987.

[36] C. P. Schnorr, "Efficient identification and signatures for smart cards," in *Advances in Cryptology—Crypto'89 Proceedings*, pp. 239–252, Springer, 1990.

[37] J. Camenisch, "Direct anonymous attestation explained," tech. rep., IBM Research, July 2007.

[38] IBM Research Zürich Security team, "Specification of the Identity Mixer cryptographic library, version 2.3.4," tech. rep., IBM Research, Zürich, February 2012.

[39] D. Pointcheval, "The composite discrete logarithm and secure authentication," in *Public Key Cryptography*, pp. 113–128, Springer, 2000.

[40] P. Vullers, "Irma/idemix proof generation & verification." (unpublished), September 2013.

[41] W. L. Gergely Alpár, Jaap-Henk Hoepman and P. Vullers, "Fast Revocation of Attribute-Based Credentialsfor Users and Verifiers." (unpublished), April 2014.

[42] E. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," in *Proceedings of the 11th ACM conference on Computer and communications security*, pp. 132–145, ACM, 2004.

[43] G. Alpár and J.-H. Hoepman, "A secure channel for attribute-based credentials," *ACM Digital Identity Management Workshop (DIM)*, pp. 13–18, 2013.