BRNO UNIVERSITY OF TECHNOLOGY

Faculty of Electrical Engineering
and Communication

DOCTORAL THESIS

Brno, 2019 Ing. Petr Dzurenda

BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF TELECOMMUNICATIONS
ÚSTAV TELEKOMUNIKACÍ

CRYPTOGRAPHIC PROTECTION OF DIGITAL IDENTITY
KRYPTOGRAFICKÁ OCHRANA DIGITÁLNÍ IDENTITY

DOCTORAL THESIS
DIZERTAČNÍ PRÁCE

AUTHOR — **Ing. Petr Dzurenda**
AUTOR PRÁCE

SUPERVISOR — **doc. Ing. Jan Hajný, Ph.D.**
ŠKOLITEL

BRNO 2019

ABSTRACT

The doctoral thesis deals with privacy-preserving cryptographic schemes in access control and data collection areas. Currently, card-based physical access control systems are used by most people on a daily basis, for example, at work, in public transportation and at hotels. However, these systems have often very poor cryptographic protection. For instance, user identifiers and keys can be easily eavesdropped and counterfeited. Furthermore, privacy-preserving features are almost missing and, therefore, user's movement and behavior can by easily tracked. Service providers (and even eavesdroppers) can profile users, know what they do, where they go, and what they are interested in. In order to improve this state, we propose four novel cryptographic schemes based on efficient zero-knowledge proofs and elliptic curve cryptography. In particular, the thesis presents three novel privacy-friendly authentication schemes for access control and one for data collection application scenarios. The first scheme supports distributed multi-device authentication with multiple Radio-Frequency IDentification (RFID) user's devices. This feature is particularly important in applications for controlling access to dangerous areas where the presence of protective equipment is checked during each access control session. The other two presented schemes use attribute-based approach to protect user's privacy, i.e. these schemes allow users to anonymously prove the ownership of their attributes, such as age, citizenship, and gender. While one of our scheme brings efficient revocation and identification mechanisms, the other one provides the fastest authentication phase among the current state of the art solutions. The last (fourth) proposed scheme is a novel short group signature scheme for data collection scenarios. Data collection schemes are used for secure and reliable data transfer from multiple remote nodes to a central unit. With the increasing importance of smart meters in energy distribution, smart house installations and various sensor networks, the need for secure data collection schemes becomes very urgent. Such schemes must provide standard security features, such as confidentiality and authenticity of transferred data, as well as novel features, such as strong protection of user's privacy and identification of malicious users. The proposed schemes are provably secure and provide the full set of privacy-enhancing features, namely anonymity, untraceability and unlinkability of users. Besides the full cryptographic specification and security analysis, we also show the results of our implementations on devices commonly used in access control and data collection applications.

KEYWORDS

Cryptography, Privacy, Group Signatures, Attribute-Based Credentials, Anonymity, Smart Cards, Authentication, Elliptic Curves, Bilinear Pairing, Constrained Devices

ABSTRAKT

Dizertační práce se zabývá kryptografickými schématy zvyšující ochranu soukromí uživatelů v systémech řízení přístupu a sběru dat. V současnosti jsou systémy fyzického řízení přístupu na bázi čipových karet využívány téměř dennodenně většinou z nás, například v zaměstnání, ve veřejné dopravě a v hotelech. Tyto systémy však stále neposkytují dostatečnou kryptografickou ochranu a tedy bezpečnost. Uživatelské identifikátory a klíče lze snadno odposlechnout a padělat. Funkce, které by zajišťovaly ochranu soukromí uživatele, téměř vždy chybí. Proto je zde reálné riziko možného sledovaní lidí, jejich pohybu a chovaní. Poskytovatelé služeb nebo případní útočníci, kteří odposlouchávají komunikaci, mohou vytvářet profily uživatelů, ví, co dělají, kde se pohybují a o co se zajímají. Za účelem zlepšení tohoto stavu jsme navrhli čtyři nová kryptografická schémata založená na efektivních důkazech s nulovou znalostí a kryptografii eliptických křivek. Konkrétně dizertační práce prezentuje tři nová autentizační schémata pro využití v systémech řízení přístupu a jedno nové schéma pro využití v systémech sběru dat. První schéma využívá distribuovaný autentizační přístup vyžadující spolupráci více RFID prvků v autentizačním procesu. Tato vlastnost je výhodná zvláště v případech řízení přístupu do nebezpečných prostor, kdy pro povolení přístupu uživatele je nezbytné, aby byl uživatel vybaven ochrannými pomůckami (se zabudovanými RFID prvky). Další dvě schémata jsou založena na atributovém způsobu ověření, tj. schémata umožňují anonymně prokázat vlastnictví atributů uživatele, jako je věk, občanství a pohlaví. Zatím co jedno schéma implementuje efektivní revokační a identifikační mechanismy, druhé schéma poskytuje nejrychlejší verifikaci držení uživatelských atributů ze všech současných řešení. Poslední, čtvrté schéma reprezentuje schéma krátkého skupinového podpisu pro scénář sběru dat. Schémata sběru dat se používají pro bezpečný a spolehlivý přenos dat ze vzdálených uzlů do řídící jednotky. S rostoucím významem chytrých měřičů v energetice, inteligentních zařízení v domácnostech a rozličných senzorových sítí, se potřeba bezpečných systémů sběru dat stává velmi naléhavou. Tato schémata musí podporovat nejen standardní bezpečnostní funkce, jako je důvěrnost a autentičnost přenášených dat, ale také funkce nové, jako je silná ochrana soukromí a identity uživatele či identifikace škodlivých uživatelů. Navržená schémata jsou prokazatelně bezpečná a nabízí celou řadu funkcí rozšiřující ochranu soukromí a identity uživatele, jmenovitě se pak jedná o zajištění anonymity, nesledovatelnosti a nespojitelnosti jednotlivých relací uživatele. Kromě úplné kryptografické specifikace a bezpečnostní analýzy navržených schémat, obsahuje tato práce také výsledky měření implementací jednotlivých schémat na v současnosti nejpoužívanějších zařízeních v oblasti řízení přístupu a sběru dat.

KLÍČOVÁ SLOVA

Kryptografie, Soukromí, Skupinové Podpisy, Atributová Pověření, Anonymita, Čipové Karty, Autentizace, Eliptické Křivky, Bilineární Párování, Omezená Zařízení

DZURENDA, Petr. *Cryptographic protection of digital identity*. Brno, 2019, 159 p. Doctoral thesis. Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications. Advised by Assoc. prof. Jan Hajný, Ph.D.

DECLARATION

I declare that I have written the Doctoral Thesis titled "Cryptographic protection of digital identity" independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the thesis and listed in the comprehensive bibliography at the end of the thesis.

As the author I furthermore declare that, with respect to the creation of this Doctoral Thesis, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll., Section 2, Head VI, Part 4.

Brno ....

author's signature

ACKNOWLEDGEMENT

I would like to thank my supervisor Assoc. prof. Jan Hajný, Ph.D. for the guidance and professional support during the work on this thesis. I am grateful to Dr. Jan Camenisch and Dr. Manu Drijvers from IBM Research in Zurich (Switzerland), Dr. Jordi Castellà-Roca from Universitat Rovira i Virgili in Tarragona (Spain/Catalonia) and Dr. Lukáš Malina and Assoc. prof. Václav Zeman, Ph.D. from Brno University of Technology in Brno (Czech Republic) for cooperation and common research in fields of privacy-enhancing technologies. I would like to express gratitude to my parents Ján and Irena Dzurendovi, my brother Ján, and my dearest fiancée Sara Ricci, to my colleagues and all my friends for their patience, understanding and their unconditional support.

Brno ....

author's signature

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
http://www.six.feec.vutbr.cz

ACKNOWLEDGEMENT

Research described in this Doctoral Thesis has been implemented in the laboratories supported by the SIX project; reg. no. CZ.1.05/2.1.00/03.0072, operational program Výzkum a vývoj pro inovace.

Brno ....

author's signature

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

To my parents, my brother and Sara for their infinite support and help. Thank you for being here for me.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

We live in the Information Age. The time when the ownership of a computer or the Internet access was just a privilege for rich people only has already gone. Current "*smart*" devices are permanently connected to the Internet and provide us a great deal of different cloud services. Smart devices, the Internet and many of cloud solutions form our daily life. The Internet is no longer used just to search for information. For example, new smart televisions (TV) allow us to watch on-line streaming videos and record movies which are stored to the cloud. Smart phones are not used just to make calls. For instance, they can be used for sport activities (as a personal trainer, e.g. Endomondo), listening to music (on-line streaming music services such as Spotify), chatting with friends, and living our social life (on Facebook, Google, WhatsApp etc.). Our data are always available thanks to services such as Dropbox, Google Drive, OneDrive, and iCloud. Moreover, we never get lost, since our smart phone is equipped with Global Positioning System (GPS). At present, there is almost nobody who misses an account held under Facebook, Google, Amazon or Apple. These internet giants collect our data and profile us. They may do that for improving and optimising the services or for better understanding our behaviour and preferences [3]. But how can we be sure that these data are not collected to track us and sell our profile?

Smart grids, smart metering or smart cities are the current terms as well as the Internet of Things (IoT). Electronic devices start to communicate with each other without human interaction, they send (or exchange) many of user data through the Internet. New data published by Juniper Research [4] show that the development of smart grids linked to the smart cities will result in citizens saving $14 billion per annum in energy bills by 2022. Most of the big cities (such as London, Brussels, Barcelona and many others) apply the low emission zones at the city centers to minimize the pollution. In this scenario, only registered cars have access to the center. The bicycle– , scooter– or even car–sharing is an actual service in the most modern cities. In many cases, just a pre-installed application in a smart phone with Bluetooth or Near Field Communication (NFC) technology support is required to unlock and use these vehicles. The public transportation system gets more and more integrated, and, at the same time, supports smart cards with prepaid fare. Countries issue electronic IDentity (eID) cards as in the case of Germany [5] and Czechia [6]. These may lead to people tracking anywhere at any time.

The current systems are required to provide standard security properties. The data has to be protected against modification (data *integrity*) and eavesdropping

(data *confidentiality*). The data recipient has to be sure that the data was sent by a known sender (*authentication*) and the sender cannot deny having sent the data (*non-repudiation*). Unfortunately, the standard systems use the identity-based authentication approach, where a user must identity himself at first. To do that, he sends his unique identifier (which is associated with his real identity), and then, he proves the proclaimed identity using the corresponding private key. This security context has a big impact on user's privacy, since the user identity is always disclosed. The verifier or service provider can profile the user, track his movement and behavior. Therefore, the standard security requirements are insufficient. In many scenarios user identification is not necessary and a service provider needs to know only whether a user has access to the required service (i.e., holds a valid ticket) or not. No other personal information is needed. The requirements on development of more privacy-friendly applications have been already demanded since 2011 by United States (US) [7] and European Union (EU) [8] institutions.

Especially recently, the European Commission has adopted many new regulations and strategies with close relation to the user privacy. For example, the General Data Protection Regulation (GDPR) [9] is the regulation of EU law from 2016. In particular, GDPR aims primarily on data protection and privacy. Thanks to this regulation, users gain higher control over their data. The European Network and Information Security Agency (ENISA) demands on privacy-preserving features of European eID [10]. The European Strategy on Cooperative Intelligent Transport Systems (C-ITS) [11] aims to improve road safety, traffic efficiency and comfort of driving, by helping drivers to take the right decisions and adapt their route to the traffic situation. In this context, C-ITS assumes that there is a communication between vehicles and a transport infrastructure. Drivers are exchanging information about their locations and other important data. In the same time, C-ITS must protect the location privacy of drivers to avoid their tracking.

Modern cryptographic constructions may prevent privacy leaks in current scenarios. For example, group and ring signatures significantly increase user's privacy. Users only prove their membership in the specific group, while their identity remains hidden. Furthermore, Attribute-Based Credential (ABC) schemes allow users to prove the possession of personal attributes, while no more additional information or user's identity is revealed. Therefore, these schemes are suitable for privacy-friendly systems. Unfortunately, the schemes are usually more computationally expensive compared to standard signature and authentication schemes, since they use more arithmetic operations. In particular, the modular exponentiation and bilinear pairing operations are widely used and directly affect the scheme efficiency, and, hence, its practical usability.

1.2 Thesis Objectives

The general objective of this thesis is to design novel privacy-enhancing cryptographic schemes for practical use in current Information and Communication Technology (ICT) scenarios, especially in access control systems, but also in data collection and notification systems. The current systems use identity-based authentication (and authorization) approaches to control the access to services. This affects directly user privacy and digital identity protection. Therefore, we are mainly interested in developing novel privacy-friendly cryptographic schemes that address these shortcomings and threats. First of all, we require that the scheme provides both *security* and *privacy* properties. The scheme must by **provably secure**, i.e., the scheme security holds under cryptographic hardness assumptions, and meets both **completeness** and **soundness** properties. Furthermore, we are going to involve advanced cryptographic primitives, such as **zero-knowledge** protocols, to control the amount of released sensitive information during the authentication process. Besides the *security* properties, the scheme has to meet at least the following *privacy* properties:

- **Anonymity**: the user's identity remains hidden during the authentication process. Hence, there is no privacy threats for honest users. The verifier may only check, whether the user is authorized to access the service or not.
- **Unlinkability**: all transactions (sessions) of a single user are mutually unlinkable and completely indistinguishable from the transactions of other users. It prevents linking individual sessions together and profiling users.
- **Untraceability**: the proofs generated by users are randomized, hence, not even the issuer is able to track issued credentials, i.e., users' behaviour or movement.

Moreover, we require that the scheme provides efficient **revocation** and **identification** mechanisms. This allows a service provider to learn the user's identity in case of malicious intents. If the user loses his credentials (typically a smart card with stored user secret keys and relevant attributes), the service provider can revoke the user from the system, by putting the user revocation handlers on the blacklist.

Most current scenarios involve many constrained devices (wearables, smart meters, sensors, RFID tags, smart cards etc.) with computation and memory limitations. Accordingly, we require the scheme to be **sufficiently fast even on constrained devices, in particular on smart cards**. Smart cards are considered to be tamper-resistant devices and, therefore, they provide secure storage for sensitive data, including user private keys. For this reason, we design novel schemes based on **elliptic curve cryptography** to reduce computational and memory resources

on smart cards. It is important to notice that some operations (such as bilinear pairing) cannot be used on smart cards due to their unavailability on these devices.

1.3 Thesis Structure

The thesis is organized as follows: Chapter 2 defines cryptographic preliminaries. The used notation, hardness assumptions, elliptic curve cryptography issues, and cryptographic primitives that are used in our privacy-enhancing schemes are defined here. Chapter 3 presents a comprehensive state of the art analysis of the current privacy-enhancing schemes. In particular, group signatures and attribute-based credential schemes are presented and compared. Moreover, we provide state of the art of the current smart cards focusing on elliptic curve cryptography support and benchmarks. Chapter 4 introduces a novel multi-device authentication scheme with strong privacy protection. A novel short group signature scheme is presented in Chapter 5. Novel attribute-based credential schemes are presented in Chapter 6 and Chapter 7. The scheme presented in Chapter 6 includes revocation mechanisms and provides non-repudiation properties, while the scheme in Chapter 7 is the fastest one from the current solutions. The thesis conclusions are presented in Chapter 8.

2 Cryptographic Preliminaries

This section contains cryptographic preliminaries which are related with proposed schemes in Chapter 4, 5, 6, and 7. The content of this chapter comes from the published papers [12], [13], [14], [15], [16].

2.1 Notation

We use the notation introduced by Camenisch and Stadler (CS) [17] to describe Proof of Knowledge (PK) protocols. Let c be a number in a finite group $\mathbb{K}$ and g be a generator of the same group $\mathbb{K}$. The protocol proving the knowledge of discrete logarithm w of c with respect to g is denoted as $PK\{w : c = g^w\}$. Equivalently, given C, G two points of an elliptic curve E over a finite field $\mathbb{F}_p$, where G is a base point of E, the protocol proving the knowledge of Elliptic Curve (EC) discrete logarithm of C with respect to G is denoted as $PK\{w : C = w \bullet G\}$. Furthermore, we use the proof of representation denoted as $PK\{w_0, w_1, \ldots, w_i : c = g^{w_0} g^{w_1} \cdots g^{w_i}\}$ in the standard variant and as $PK\{w_0, w_1, \ldots, w_i : C = w_0 \bullet G_0 + w_1 \bullet G_1 + \cdots + w_i \bullet G_i\}$ in the EC variant. The proof of discrete logarithm equivalence with respect to different generators $g_1,\ g_2 \in \mathbb{K}$ is denoted as $PK\{w : c_1 = g_1^w \wedge c_2 = g_2^w\}$. A signature by a traditional scheme (e.g., RSA) of an entity $\mathcal{E}$ on some data is denoted as $Sig_{\mathcal{E}}(data)$.

The symbol "$\cdot$" denotes multiplication, "$\bullet$" denotes scalar EC point multiplication (the notation is primarily used in Chapter 6 to easily distinguish between finite group $\mathbb{K}$ and group generated by $E(\mathbb{F}_p)$), ":" means "such that", "|" means "divides", "$|x|$" is the bitlength of x, "$x \xleftarrow{\$} \{0,1\}^l$" is a randomly chosen bitstring of maximum length l and we write $a \xleftarrow{\$} A$ when a is sampled uniformly at random from A. We write $\mathbb{G} = \langle g \rangle$ when g generates the group $\mathbb{G}$. A secure hash function is denoted as $\mathcal{H}$. In Section 2.2, the notation $\mathcal{A} \leq_p \mathcal{B}$ means that problem $\mathcal{A}$ is polynomial time reducible to problem $\mathcal{B}$ and the symbol $\mathcal{O}$ denotes a random oracle. We denote the set of all issued attributes as $\mathcal{A}$, while the set of disclosed attributes is denoted as $\mathcal{D}$.

2.2 Hardness Assumptions

In public key cryptography, deriving the private key from the public key is considered a hard problem. It is made possible thanks to the use of *computational hardness assumptions* on the related problems. The assumption is a hypothesis that a particular (cryptographic) problem cannot be solved efficiently, i.e. in polynomial time. This hypothesis allows us to design cryptographic primitives, that are provable secure. Therefore, any cryptographic scheme based on these primitives is considered

to be secure against computationally bounded adversaries $\mathcal{A}$, that are running in Probabilistic Polynomial Time (PPT), as all real adversaries $\mathcal{A}$ actually are.

There are plenty of hardness assumptions used in cryptography, see [18]. However, most of the current cryptographic schemes are based on assumptions related to Integer Factorization (IF) problem and Discrete Logarithm (DL) problem.

Integer Factorization Problem

IF problem is related to hardness of finding prime factors $p_1, p_2, \ldots, p_k$ of a given large composite number $N \in \mathbb{N}$, such that $N = \prod_{i=1}^{k} p_i^{e_i}$, where $e_i \geq 1$. While a multiplication is *easy* to compute, finding the prime factorization of a large number in PPT is generally considered to be hard. For simplicity, we assume N to be Rivest, Shamir and Adleman (RSA) modulus, i.e. $N = pq$, where p, q are large κ-bit prime numbers, see Assumption 1. This assumption is used to provide provable security of the RSA cryptosystem [19]. For security reasons, the National Institute of Standards and Technology (NIST) [2] claims that modulus length of at least 2048 bits should be used. This request is related with $\kappa = 1024$ bits (bitlength of each prime p, q).

Assumption 1 (**Integer Factorization**). *Let $\mathcal{O}^{\mathtt{IF}}(\cdot)$ on input $p, q \in \{0,1\}^{\kappa} \bigcap \mathbb{P}$ outputs $N \in \mathbb{N}$. Define the advantage of an adversary $\mathcal{A}$ as follows,*

$$\mathsf{Adv}_{\mathsf{IF}}(\mathcal{A}) = \mathsf{Pr}\Big[(N,p,q) \leftarrow \mathit{GenModulus}(1^{\kappa}), (p,q) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathtt{IF}}(\cdot)}(N) : pq = N\Big].$$

No PPT adversary has $\mathsf{Adv}_{\mathsf{IF}}(\mathcal{A})$ non-negligible in κ.

RSA assumption is another popular assumption, which is a potentially stronger than hardness factorisation assumption. Moreover, it is proven that RSA $\leq_p$ IF. The assumption is stemmed from RSA cryptosystem [19]. RSA assumption implies that factoring is hard, if the inverse of the public exponent e is not known. In other words, the RSA problem [20] is to find the plaintext $m \in \mathbb{Z}_n^*$, such that $c \equiv m^e \pmod{N}$, for given RSA public key (modulus $N = pq$, exponent e), and ciphertext c, see Assumption 2 for more details. A well-known algorithm based on the RSA assumption is RSA cryptosystem.

Assumption 2 (**RSA**). *Let $\mathcal{O}_e^{\mathtt{RSA}}(\cdot)$ on input $m \in \mathbb{Z}_N^*$ outputs $m^e \bmod N$. Define the advantage of an adversary $\mathcal{A}$ as follows,*

$$\mathsf{Adv}_{\mathsf{RSA}}(\mathcal{A}) = \mathsf{Pr}\Big[(N,e,d) \leftarrow \mathit{GenRSA}(1^{\kappa}), y \xleftarrow{\$} \mathbb{Z}_N^*, x \leftarrow \mathcal{A}^{\mathcal{O}_e^{\mathtt{RSA}}(\cdot)}(N,y) : y \equiv x^e \pmod{N}\Big].$$

No PPT adversary has $\mathsf{Adv}_{\mathsf{RSA}}(\mathcal{A})$ non-negligible in κ.

Strong RSA (SRSA) assumption is stronger than the RSA assumption, since adversary $\mathcal{A}$ can additionally choose the public exponent $e \geq 3$. More specifically, Strong RSA problem is to find the pair (m, e) such that $c \equiv m^e \pmod{N}$ for given RSA modulus N and ciphertext c, see Assumption 3. The appropriate reduction is SRSA $\leq_p$ RSA. The representative of the scheme that relies on the Strong RSA assumption is Idemix [21] (attribute-based credential scheme, see more details in Section 3.2.2).

Assumption 3 (**Strong RSA**). *Let* $\mathcal{O}^{\mathtt{RSA}}(\cdot)$ *on input* $m \in \mathbb{Z}_N^*$ *and* $e \geq 3$ *outputs* $m^e \bmod N$. *Define the advantage of an adversary* $\mathcal{A}$ *as follows,*

$$\mathsf{Adv}_{\mathsf{SRSA}}(\mathcal{A}) = \mathsf{Pr}\Big[(N,p,q) \leftarrow \texttt{GenModulus}(1^\kappa), y \xleftarrow{\$} \mathbb{Z}_N^*, \\ (x,e) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathtt{RSA}}(\cdot)}(N,y) : y \equiv x^e \pmod{N} \wedge e \geq 3\Big].$$

No PPT adversary has $\mathsf{Adv}_{\mathsf{SRSA}}(\mathcal{A})$ *non-negligible in* κ.

Discrete Logarithm Problem

DL problem is defined in any finite cyclic group $\mathbb{G}$, typically it is used in (1) the multiplicative group over composite number $\mathbb{Z}_n^*$ (RSA modulus $n = pq$), (2) over the prime number $\mathbb{Z}_p^*$, or (3) the group generated by elliptic curve over finite field $E(\mathbb{F}_{q^m})$. In our case, we consider the group $\mathbb{G} = \langle g \rangle$ of prime order q, where $|q| = \kappa$. The discrete logarithm problem is about computing x from given group elements $g, h \in \mathbb{G}$, such that $h = g^x$, see Assumption 4. Assumption 4 is used in Digital Signature Algorithm (DSA) scheme [22]. For security reasons, NIST [2] claims that the modulus length of at least 2048 bits should be used.

Assumption 4 (**Discrete Logarithm**). *Let* $\mathcal{O}_x^{\mathtt{DL}}(\cdot)$ *on input* $g \in \mathbb{G}$ *outputs* g^x. *Define the advantage of an adversary* $\mathcal{A}$ *as follows,*

$$\mathsf{Adv}_{\mathsf{DL}}(\mathcal{A}) = \mathsf{Pr}\Big[(\mathbb{G}, g, q) \leftarrow \texttt{GroupSetup}(1^\kappa), h \xleftarrow{\$} \mathbb{G}, x \leftarrow \mathcal{A}^{\mathcal{O}_x^{\mathtt{DL}}(\cdot)}(h) : h = g^x\Big].$$

No PPT adversary has $\mathsf{Adv}_{\mathsf{DL}}(\mathcal{A})$ *non-negligible in* κ.

Many assumptions imply DL problem together with some problems related with Diffi-Hellman (DH) protocol [23]. Namely, Computational Diffie-Hellman (CDH) assumption supposes that it is hard for given triplet (g, g^a, g^b) for unknown $a, b \in \mathbb{Z}_q$ to compute $c = g^{ab}$, see Assumption 5. The assumption is potentially stronger than DL assumption, since the problem is reducible CDH $\leq_p$ DL. In cryptography, it is used in Diffie-Hellman key exchange protocol [23] and ElGamal cryptosystem [24].

Assumption 5 (**Computational DH**). *Let* $\mathcal{O}^{\mathtt{DH}}_{a,b}(\cdot)$ *on input* $g \in \mathbb{G}$ *outputs* g^{ab}. *Define the advantage of an adversary* $\mathcal{A}$ *as follows,*

$$\mathsf{Adv}_{\mathsf{CDH}}(\mathcal{A}) = \mathsf{Pr}\Big[(\mathbb{G}, g, q) \leftarrow \mathtt{GroupSetup}(1^\kappa), \\ (a, b) \xleftarrow{\$} \mathbb{Z}_q, c \leftarrow \mathcal{A}^{\mathcal{O}^{\mathtt{DH}}_{a,b}(\cdot)}(g, g^a, g^b) : c = g^{ab}\Big].$$

No PPT adversary has $\mathsf{Adv}_{\mathsf{CDH}}(\mathcal{A})$ *non-negligible in* κ.

Static DH (SDH) problem supposes that computing h^a is hard for fixed values $g, g^a \in \mathbb{G}$ and given element $h \in \mathbb{G}$, see Assumption 6. The assumption is potentially stronger than CDH assumption, since the problem is reducible SDH $\leq_p$ CDH.

Assumption 6 (**Static DH**). *Let* $\mathcal{O}^{\mathtt{DH}}_{a}(\cdot)$ *on input* $h \in \mathbb{G}$ *outputs* h^a. *Define the advantage of an adversary* $\mathcal{A}$ *as follows,*

$$\mathsf{Adv}_{\mathsf{SDH}}(\mathcal{A}) = \mathsf{Pr}\Big[(\mathbb{G}, g, q) \leftarrow \mathtt{GroupSetup}(1^\kappa), \\ a \xleftarrow{\$} \mathbb{Z}_q, a' \leftarrow \mathcal{A}^{\mathcal{O}^{\mathtt{DH}}_{a}(\cdot)}(g, g^a) : a' = a\Big].$$

No PPT adversary has $\mathsf{Adv}_{\mathsf{SDH}}(\mathcal{A})$ *non-negligible in* κ.

Decision DH (DDH) assumption is another potentially stronger assumption than CDH assumption. DDH assumption supposes that for given triplet $h, g^a, g^b \in \mathbb{G}$ for unknown $a, b \in \mathbb{Z}_q$, it is hard to determine, whether or not $h = g^{ab}$. The problem is reducible DDH $\leq_p$ CDH. In cryptography, DDH assumption is used in Diffie-Hellman key exchange protocol [23] and ElGamal cryptosystem [24].

Assumption 7 (**Decision DH**). *Let* $\mathcal{O}^{\mathtt{DH}}_{a,b}(\cdot)$ *on input* $g \in \mathbb{G}$ *outputs* g^{ab}. *Define the advantage of an adversary* $\mathcal{A}$ *as follows,*

$$\mathsf{Adv}_{\mathsf{DDH}}(\mathcal{A}) = \mathsf{Pr}\Big[(\mathbb{G}, g, q) \leftarrow \mathtt{GroupSetup}(1^\kappa), (a, b) \xleftarrow{\$} \mathbb{Z}_q, \\ y_0 \leftarrow g^{ab}, y_1 \xleftarrow{\$} \mathbb{G}, \beta \leftarrow \{0,1\}, \beta' \leftarrow \mathcal{A}^{\mathcal{O}^{\mathtt{DH}}_{a,b}(\cdot)}(g^a, g^b, y_\beta) : \beta = \beta'\Big].$$

No PPT adversary has $\mathsf{Adv}_{\mathsf{DDH}}(\mathcal{A})$ *non-negligible in* κ.

Strong DDH (SDDH) assumption is stronger than DDH assumption. Given elements $g, g^a, g^b, g^{b^{-1}}, h \in \mathbb{G}$ for unknown $a, b, b^{-1} \in \mathbb{Z}^*_q$, it is hard to determinate, whether or not $h = g^{ab}$, even if adversary $\mathcal{A}$ knows g and $g^{b^{-1}}$. The problem is reducible SDDH $\leq_p$ DDH. The assumption is used in Boneh-Boyen signatures [25].

Assumption 8 (**Strong DDH**). *Let* $\mathcal{O}^{\mathtt{DH}}_{a,b}(\cdot)$ *on input* $g \in \mathbb{G}$ *outputs* g^{ab}. *Define the advantage of an adversary* $\mathcal{A}$ *as follows,*

$$\mathsf{Adv}_{\mathsf{SDDH}}(\mathcal{A}) = \mathsf{Pr}\Big[(\mathbb{G}, g, q) \leftarrow \texttt{GroupSetup}(1^\kappa), (a, b) \xleftarrow{\$} \mathbb{Z}_q^*, \\ y_0 \leftarrow g^{ab}, y_1 \xleftarrow{\$} \mathbb{G}, \beta \leftarrow \{0,1\}, \beta' \leftarrow \mathcal{A}^{\mathcal{O}^{\mathtt{DH}}_{a,b}(\cdot)}(g, g^a, g^b, g^{b^{-1}}, y_\beta) : \beta = \beta'\Big].$$

No PPT adversary has $\mathsf{Adv}_{\mathsf{SDH}}(\mathcal{A})$ *non-negligible in* κ.

SDDH Inversion (SDDHI) assumption is even more stronger than SDDH assumption. The best known algorithm to break SDDHI assumption is to solve DL problem. Hence, the problem is reducible SDDHI $\leq_p$ DL. The assumption is used in Boneh-Boyen signatures [25].

Assumption 9 (**SDDH Inversion**). *Let* $\mathcal{O}^{\mathtt{DH}}_{a}(\cdot)$ *on input* $z \in \mathbb{Z}_q^*$ *outputs* $g^{1/(a+z)}$. *Define the advantage of an adversary* $\mathcal{A}$ *as follows,*

$$\mathsf{Adv}_{\mathsf{SDDHI}}(\mathcal{A}) = \mathsf{Pr}\Big[(\mathbb{G}, g, q) \leftarrow \texttt{GroupSetup}(1^\kappa), a \xleftarrow{\$} \mathbb{Z}_q^*, (x, \alpha) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathtt{DH}}_{a}(\cdot)}(g, g^a) \\ y_0 \leftarrow g^{1/(a+x)}, y_1 \xleftarrow{\$} \mathbb{G}, \beta \leftarrow \{0,1\}, \beta' \leftarrow \mathcal{A}^{\mathcal{O}^{\mathtt{DH}}_{a}(\cdot)}(y_\beta, \alpha) : \beta = \beta'\Big].$$

No PPT adversary has $\mathsf{Adv}_{\mathsf{SDDHI}}(\mathcal{A})$ *non-negligible in* κ.

2.3 Elliptic Curve Cryptography

In the previous section we introduced the most common hardness assumptions, which are mostly related to IF and DL problems in finite cyclic groups $\mathbb{G}$, such as $\mathbb{Z}_N^*$ (composite order) and $\mathbb{Z}_p^*$ (prime order). However, Discrete Logarithm Problem (DLP) can by applied to any finite cyclic group $\mathbb{G}$. In fact, currently the most popular group $\mathbb{G}$ is the one generated by an elliptic curve E over a finite field $\mathbb{F}_q$, see Section 2.3.1. In this context, we speak about Elliptic Curve Discrete Logarithm (ECDL) problem. ECDL problem is harder to solve than classical DL problem, since the currently fastest known algorithm for ECDL problem solving has a full exponential cost, instead of sub-exponential in case of the classical DL problem variant. The contributions in this section have been published in scientific papers [12] and [26].

Elliptic Curve Cryptography (ECC) provides comparable or higher level of security strength than IF or DL cryptography at much smaller key sizes, i.e. it offers improved security with reduced computational requirements. In fact, *traditional* structures constructed over RSA groups [19] or DSA groups [22], require elements of at least 2048 bits long. On the other hand, elliptic curves have elements almost

10 times smaller. With the increasing size of security parameters, the difference becomes even larger. According to the NIST recommendations [2] (for the period of 2016 - 2030), the current structures should be at least 2048-bit long for *traditional* structures and 224-bit long for EC-based structures, see Table 2.1. Even in practical applications, the EC variants of protocols replaced the older schemes, especially in the case of signature schemes, encryption and key agreement schemes. We briefly review some of them.

Tab. 2.1: Key recommendation for public key cryptosystems by NIST agency.

Date	**Minimum of Strength**	**Discrete Logarithm**		**Factoring**	**Elliptic**
		Modulus	**Key**	**Modulus**	**Curve**
(Legacy)	80	1024	160	1024	160
2016-2030	112	2048	224	2048	224
2016-Beyond	128	3072	256	3072	256
2016-Beyond	192	7680	384	7680	384
2016-Beyond	256	15360	512	15360	512

Note: The key sizes are represented as bitlength and indicate the minimal sizes for the given security strengths.

- *Elliptic Curve Digital Signature Algorithm* (ECDSA) is the elliptic curve variant of DSA and allows a user to sign a message using his private key and a verifier to verify the user signature using the user's public key, see [22] for more details. ECDSA requires, compared to DSA, smaller cryptographic keys to achieve same security level, for example 224-bit ECDSA key is comparable to 2048-bit DSA parameters. Furthermore, ECDSA generates smaller signatures, for example the ECDSA 224-bit signature scheme generates 56-byte signatures instead of 256-byte signatures generated by the RSA 2048-bit signature scheme on the same security level.

- *Elliptic Curve Diffie-Hellman* (ECDH) scheme is a key agreement protocol that allows two parties, each having an elliptic curve public/private key pair, to create a shared secret key over a insecure channel, see [27]. ECDH is the elliptic curve variant of Diffie-Hellman key agreement scheme and it is standardized by NIST SP 800-56A [28] as well as its authenticated variant Elliptic Curve Menezes-Qu-Vanstone (ECMQV). Similarly, involving EC construction, we reduce communication cost between two communicating entities from 512 bytes (in case of DH) to 56 bytes (in case of ECDH) for same 112-bit security strength.

- *Elliptic Curve Integrated Encryption Scheme* (ECIES) is a public key encryption and decryption scheme that provides data confidentiality over ECC. This scheme is a variant of the ElGamal scheme proposed by Abdalla, Bellare, and Rogaway in [29] and it is standardized by the Standards for Efficient Cryptography Group (SECG) [30]. Compared to RSA, ECIES scheme requires smaller size of keys and cryptograms (ECIES 224-bit uses symmetric cipher AES-256-bit and hash function SHA-256-bit to generate cryptograms of size 120 bytes instead of 256 bytes cryptograms of RSA 2048-bit).

2.3.1 Elliptic Curve

An elliptic curve E is an algebraic curve, which can be constructed over different fields, such as $\mathbb{R}, \mathbb{C}, \mathbb{Q}$ or $\mathbb{F}_q$. Elliptic curves E over field $\mathbb{R}$ are good for understanding basic principles, however, elliptic curves cryptography is based on elliptic curve E *over a finite field* $\mathbb{F}_q$, where $q = p^m$ with p prime and $m \geq 1$, see Figure 2.1. Since we are interested in the use of elliptic curves in cryptography, we will tackle definition and main properties of elliptic curves over finite fields here. An elliptic curve is an algebraic curve that is given by an equation of the form:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{2.1}$$

where $a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}_q$ are the coefficients of the curve and are constants. This equation is often called *generalized Weierstrass form.* Any elliptic curve can be written in this form. $E(\mathbb{F}_q)$ denotes the set of points $(x, y) \in \mathbb{F}_q^2$ that satisfies this equation, along with a "point at infinity" denoted $\mathcal{O}$. Representation of points using (x, y) is known as *affine coordinates.* If $\mathbb{F}$ is a finite field, then there are only finitely many pairs (x, y) with $x, y \in \mathbb{F}$ and the group $E(\mathbb{F})$ is finite. In order to have that the curve is an elliptic curve it must be smooth, i.e. there is no point of $E(\overline{\mathbb{F}_q})$ (let $\overline{\mathbb{F}_q}$ denote the closure of $\mathbb{F}_q$) where both the partial derivates vanish. In other words, the two equations

$$a_1y = 3x^2 + 2a_2x + a_4, \tag{2.2}$$

$$2y + a_1x + a_3 = 0 \tag{2.3}$$

cannot be simultaneously satisfied by any $(x, y) \in E(\overline{\mathbb{F}_q^2})$. Otherwise, the curve is singular with point of singularity, see Figure 2.1. In general, an elliptic curve is non-singular curve, see Figure 2.1, where the discriminant is not equal to zero, i.e. $\Delta \neq 0$. The discriminant depends on the choice of EC form. In case of the short Weierstrass curve the discriminant must satisfy the following equation:

$$y^2 = x^3 + ax + b: \quad \Delta = -16(4a^3 + 27b^2) \quad \wedge \quad \Delta \neq 0 \tag{2.4}$$

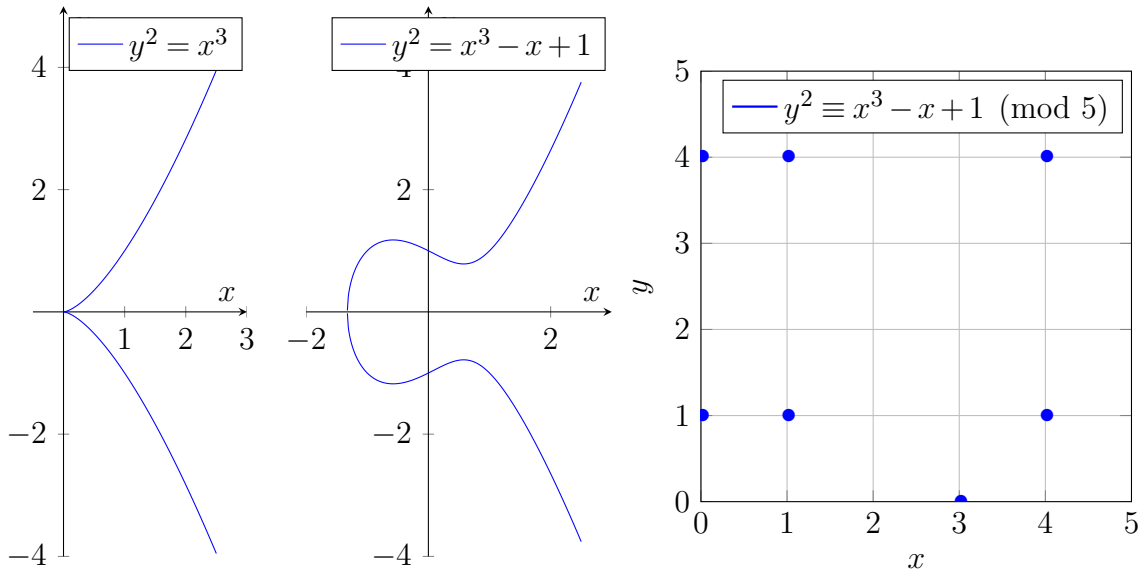

Fig. 2.1: From left to right: singular curve with a cusp singularity, non-singular curve $E(\mathbb{R})$, where $(x, y) \in \mathbb{R} \times \mathbb{R}$, and non-singular curve $E(\mathbb{F}_5)$, where $(x, y) \in \mathbb{F}_5 \times \mathbb{F}_5$.

An elliptic curve over $\mathbb{F}_p$ is called *pairing-friendly* if it contains a subgroup of order r whose embedding degree k is not too large, which means that computations in the field $\mathbb{F}_{p^k}$ are feasible. The optimal case occurs when the entire curve has prime order and the desired embedding degree. Pairing-friendly curves of prime or near-prime order are absolutely essential in certain pairing-based schemes like short signatures and group signatures.

Operations over Elliptic Curves

The points of $E(\mathbb{F}_q)$ has a group structure under an explicitly defined additive group law, i.e. given two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, them addition $R = P + Q = (x_3, y_3)$ forms a third point on the same curve. The point doubling is defined as an additional operation, where $Q = P$, and is denoted as $R = 2P$. The inverse of a point P is defined as $(x_1, -y_1)$, and the identity element is the point at infinity $\mathcal{O}$. The computation of all these elliptic curve operations is depicted in Figure 2.2.

Moreover, it is also possible to define the scalar multiple s of a point P as

$$T = sP = \underbrace{P + P + \cdots + P}_{s \text{ times}} \tag{2.5}$$

The reverse problem, i.e. computing s when only P and T are known, is intractable for carefully selected parameters. This problem is strictly related to the discrete logarithm problem, and it si called Elliptic Curve Discrete Logarithm Problem (ECDLP), which is used to provide cryptographic security. For more details see [31] or [32].

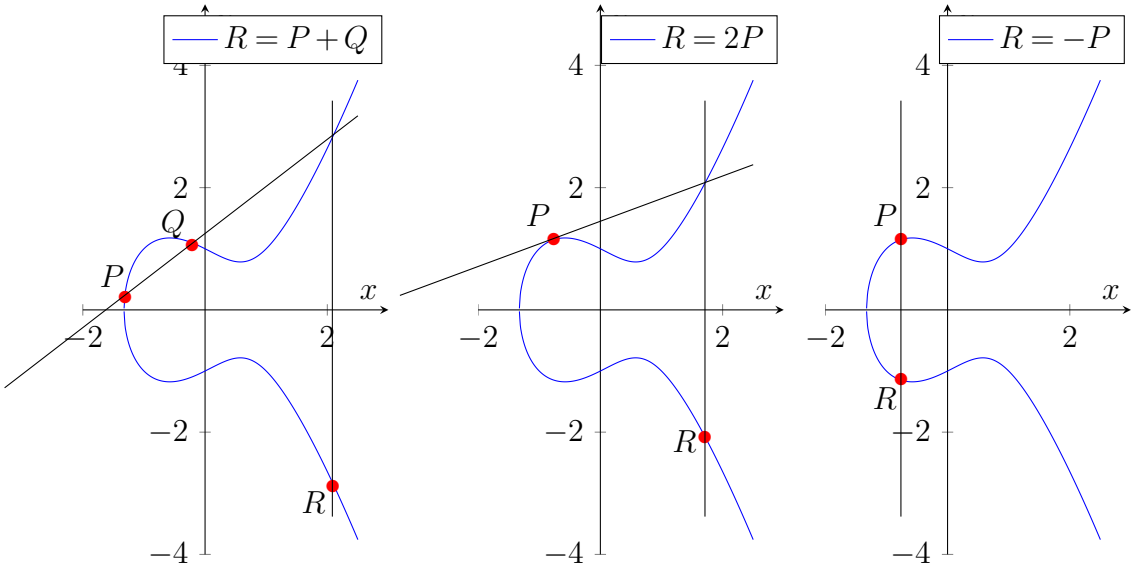

Fig. 2.2: From left to right: point addition $R = P + Q$, point doubling $R = 2P$, and point inversersion $R = -P$.

One of the main challenges in elliptic curve cryptography is to perform scalar multiplication efficiently since this operation is vital for the overall performance of the intended cryptographic algorithms. Scalar multiplication is traditionally carried out through a series of point doublings. To compute sP for a large integer s, it is inefficient to add P to itself repeatedly. It is much faster to use successive doubling. For example, to compute $19P$, we compute

$$2P, \quad 4P = 2P + 2P, \quad 8P = 4P + 4P, \quad 16P = 8P + 8P, \quad 19P = 16P + 2P + P. \tag{2.6}$$

The formulas for adding two points on an elliptic curve in Weierstrass form require 2 multiplications, 1 squaring, and 1 inversion in the field. Although finding inverses is fast, it is much slower than multiplication. In [33], it is estimated that inversion takes between 9 and 40 times more than multiplication. Moreover, squaring takes about 0.8 the time of multiplication. Often the affine representation of a curve is replaced with its form in projective coordinates since affine coordinates are expensive over prime fields due to costly field inversions.

Based on the field, we distinguish elliptic curves over prime field $E(\mathbb{F}_p)$ and binary field $E(\mathbb{F}_{2^m})$, where $m > 1$. An elliptic curve E in characteristic p is called *supersingular* if there are no points of order p, even with coordinates in an algebraically closed field. An elliptic curve E over $\mathbb{F}_q$ with order $\#E(\mathbb{F}_q) = q$ is called *anomalous curve* if the number of rational points on $\mathbb{F}_q$ is equal to the prime number q. This curve is considered weak curve because it is possible to solve ECDLP in linear time, see [34].

Elliptic Curve Form

Elliptic curves can be represented in several different forms. In order to obtain faster group operations, some curve representations are better than others. Below, we list the main forms, depicted in Figure 2.3, in which elliptic curves can be expressed.

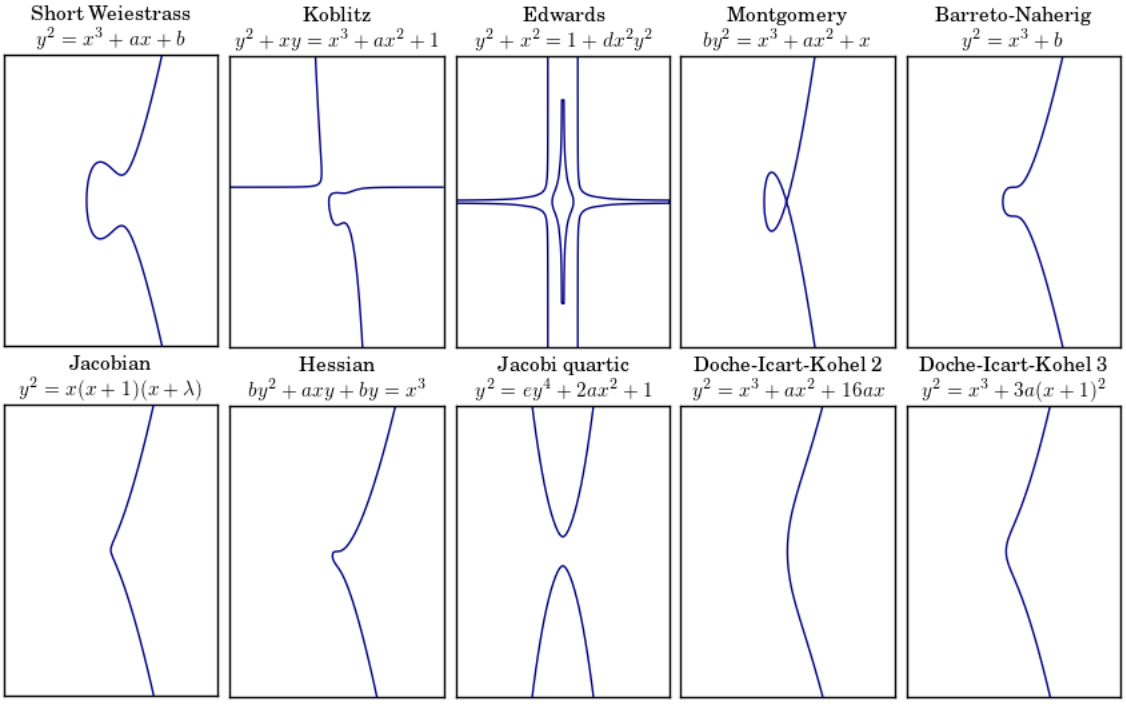

Fig. 2.3: Different elliptic curve forms in affine coordinates over $\mathbb{R}$. Jacobi intersection in affine coordinates is equal to Jacobian curve.

Short Weierstrass curve [27] is an elliptic curve with equation:

$$y^2 = x^3 + ax + b, \tag{2.7}$$

i.e. where $a_1 = a_2 = a_3 = 0$, $a_4 = a$ and $a_6 = b$ in Equation 2.1 and with $a, b \in \mathbb{F}_q$. This form can be used only in field with $p \neq 2, 3$ and, normally, it is used over prime field $\mathbb{F}_p$. Recommended secure curves are defined in many standards such as the Federal Information Processing Standards (FIPS) [22], SECG [35] Brainpool [36], the Nothing Up My Sleeve (NUMS) [37]. In FIPS [22], a is set as $a \equiv -3 \bmod p$ to achieve better efficiency of point operations.

Koblitz curve [38] has coefficients $a_1 = a_6 = 1$, $a_2 = a$ and $a_3 = a_4 = 0$ in Equation 2.1, i.e. its equation is:

$$y^2 + xy = x^3 + ax^2 + 1. \tag{2.8}$$

The curve is used in case of operations over binary fields $E(\mathbb{F}_{2^m})$ and admits especially fast elliptic scalar multiplication. The curve is also called *anomalous binary curve*. Required domain parameters are defined in FIPS [22], a takes values 0 or -1.

Edwards curve [39] is an elliptic curve with coefficient $a_1 = dxy$, $a_2 = 1$, $a_3 = a_4 = 0$, and $a_6 = 1$, i.e.,

$$x^2 + y^2 = 1 + dx^2y^2, \tag{2.9}$$

where $d \in \mathbb{F}_{p^m} \setminus \{0, 1\}$ and $p \neq 2$. Every Edwards curve is birationally equivalent to an elliptic curve with Equation 2.1, and thus admits an algebraic group law once one chooses a point to serve as a neutral element.

There exists a generalization of Edwards curves, called the *twisted Edwards curves* [40], with coefficients a and d and equation:

$$ax^2 + y^2 = 1 + dx^2y^2, \tag{2.10}$$

where $a, d \in \mathbb{F}_{p^m} \setminus \{0\}$ with $p \neq 2$. The twisted Edwards curves cover considerably more elliptic curves than Edwards curves do and even when an elliptic curve can be expressed in Edwards form, expressing the same curve in twisted Edwards form often saves time in arithmetic.

Montgomery curve [33] is described by the following equation:

$$by^2 = x^3 + ax^2 + x, \tag{2.11}$$

where $b(a^2 - 4) \neq 0$, $a \in \mathbb{F}_{p^m} \setminus \{-2, 2\}$, $b \in \mathbb{F}_{p^m} \setminus \{0\}$ and $p \neq 2$. The form is not widely used, but for example the fast curve Curve25519 [41] is widely known.

Barreto-Naehrig curve [42] is prime pairing-friendly elliptic curve over $\mathbb{F}_{p^k}$ with prime order and embedding degree $k = 12$. Barreto-Naehrig (BN) curve has equation:

$$y^2 = x^3 + b, \tag{2.12}$$

where $b \neq 0$. The curves has better speed than other known pairing-friendly elliptic curves.

Jacobian curve [43] has three points of order two defined over $\mathbb{F}_p$ with p a prime greater than 3. This means that the group order $\#E(\mathbb{F}_p)$ is divisible by 4, i.e. $\#E(\mathbb{F}_p) = 4q$ with q a prime. The equation is:

$$y^2 = x(x + 1)(x + \lambda), \tag{2.13}$$

where $\lambda \in \mathbb{F}_p$. The Jacobian form provides a defence against Simple and Differential Power Analysis (SPA/DPA) style attacks.

Jacobi curve [44] is different from the Weierstrass one and has two forms: Jacobi intersection and Jacobi quartic. These curves permit to use the same formula for the doubling and the general addition of points on the curve, and therefore, prevent SPA-like attacks on elliptic curve systems. We consider $\mathbb{F}_q$ with characteristic $p \neq 2, 3$. *Jacobi intersection* is represented as the intersection of two quadric surfaces in the projective space $\mathbb{P}^3(\mathbb{F}_q)$:

$$\begin{cases} X^2 + Y^2 - T^2 = 0 \\ (1-\lambda)X^2 + Z^2 - T^2 = 0, \end{cases} \tag{2.14}$$

where we applied the map $(x, y) \longmapsto (X, Y, Z, T) = (x, y, 1, x^2)$. In fact, any elliptic curve over $\mathbb{F}_q$ can be embedded as the intersection of two quadrics in $\mathbb{P}^3(\mathbb{F}_q)$, [45]. In affine coordinates, Equation 2.14 is equal to the Jacobian curve (Equation 2.13), see [43] for more details. Jacobi also studied quartics of the form:

$$y^2 = ex^4 + 2ax^2 + 1, \tag{2.15}$$

where normally $e = 1$. *Jacobi quartic* can be obtained from a curve in Weierstrass form (Equation 2.1) with at least one point of order 2.

Hessian curve [46] is an elliptic curve over $\mathbb{F}_q$ which has the point $(0, 0)$ of order 3. The order q has to be a prime power such that $q \equiv 2 \pmod 3$. The equation is:

$$y^2 + axy + by = x^3, \tag{2.16}$$

where $a, b \in \mathbb{F}_q$. This curve has fast elliptic curve scalar multiplication and resistance against side-channel attacks, see [47].

The speed of a curve is computed by counting the number of field multiplication (M), field squaring (S) and field multiplication by a curve constant (D) necessary for point addition and point doubling. Scalar multiplication is carried out by point addition and point doubling, therefore, it is strictly related to their speed. The standard approximations is $(S, D) \approx (0.8M, 0M)$, i.e. squaring takes about 0.8 the time of multiplication and field multiplication by a curve constant takes about 0 the time of multiplication. In our article [12], we compare different kind of elliptic curves for different approximations of squaring and doubling with respect to M. Table 2.2 shows the comparison of Short Weierstrass, Edwards, Jacobian, Jacobi quartic, Jacobi intersection and Hessian curves over $\mathbb{F}_q$, see [48] and [49] for

Tab. 2.2: Comparison of the speed of elliptic curves for point addition and point doubling over $\mathbb{F}_q$. Field multiplication is labelled by "M", field squaring by "S" and field multiplication by a curve constant by "D"; $(S, D) \approx (0.8M, 0M)$ is the standard approximation.

Elliptic Curve Form	**Addition**	$(\mathbf{0.8}, \mathbf{0})$	**Doubling**	$(\mathbf{0.8}, \mathbf{0})$
Short W. (Equation 2.7)	$12M+2S$	$16M$	$5M+6S+1D$	$9.8M$
Edwards (Equation 2.9)	$9M+1S+1D$	**9.8M**	$3M+4S$	**6.2M**
Jacobian (Equation 2.13)	$11M+5S$	$15M$	$1M+8S+1D$	$7.4M$
Jacobi q. (Equation 2.14)	$7M+3S+1D$	**9.4M**	$2M+5S+1D$	**6M**
Jacobi i. (Equation 2.15)	$11M+1S+2D$	$11.8M$	$2M+5S+1D$	$6M$
Hessian (Equation 2.16)	$6M+6S$	$10.8M$	$3M+6S$	$7.8M$

more details. Jacobi quartic and Edwards curves result the fastest for arithmetic operations from a theoretical point of view. Barreto-Naehrig curve is a particular case of Short Weierstrass curve where $a = 0$ and has operation cost equal to the Short Weierstrass one, i.e. $12M+2S$ for point addition and $5M+6S+1D$ for point doubling. Note that Barreto-Naehrig curve is over $\mathbb{F}_{p^k}$ with fixed $k = 12$ and, in [50], it is shown how to speed up the EC computations using an algorithm which merges calculations over $\mathbb{F}_{p^{12}}$ and over $\mathbb{F}_p$. In fact, if M, S are multiplication and squaring over $\mathbb{F}_{p^k}$, and m, s is multiplication and squaring over $\mathbb{F}_p$, then the curve has speed: $1M + 21m + 6s$ for point addition and $1M + 1S + 15m + 8s$ for point doubling. Koblitz curve is only used over binary fields, where the arithmetics take advantage of the field characteristic 2, e.g. the operation of addition corresponds to the eXclusive OR (XOR) operation in hardware. Furthermore, point addition and point doubling require exactly the same number of operations, that is $5M+1S$, see [33] for more details. Montgomery curve does not support fast addition, but the "Montgomery ladder" has fast scalar multiplication [48].

2.3.2 Bilinear Pairing

A *pairing* is a bilinear map from an elliptic curve group $E(\mathbb{F}_q)$ to the multiplicative group of some extension field $\mathbb{F}_{q^k}$. The parameter k is called the *embedding degree* of the elliptic curve. The embedding degree affects the security level efficiently achievable on the curve. The pairing is considered to be secure if the discrete logarithms in the groups $E(\mathbb{F}_q)$ and in $\mathbb{F}^*_{q^k}$ are both computationally infeasible. The parameters q and k should be chosen so that the two discrete logarithm problems are of approximately equal difficulty when using the best known algorithms, and the order of the group $\#E(\mathbb{F}_p)$ should have a large prime factor r. The pairings are

used to attack the discrete logarithm problem [27] and also in cryptographic setting (e.g. short signatures [51], group signatures [52] identity-based encryption [53] or identity-based signature schemes [54]).

A bilinear map is defined as follows: let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be groups of prime order q, then a bilinear map $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ must satisfy *bilinearity property*:

$$\mathbf{e}(P_1 + P_2, Q) = \mathbf{e}(P_1, Q) \cdot \mathbf{e}(P_2, Q), \quad \text{for} \quad P_1, P_2 \in \mathbb{G}_1, Q \in \mathbb{G}_2 \tag{2.17}$$

$$\mathbf{e}(P, Q_1 + Q_2) = \mathbf{e}(P, Q_1) \cdot \mathbf{e}(P, Q_2), \quad \text{for} \quad P \in \mathbb{G}_1, Q_1, Q_2 \in \mathbb{G}_2, \tag{2.18}$$

as results we get the equation

$$\mathbf{e}(P^x, Q^y) = \mathbf{e}(P, Q)^{xy}, \quad \text{for all} \quad x, y \in \mathbb{Z}_q, P \in \mathbb{G}_1, Q \in \mathbb{G}_2, \tag{2.19}$$

non-degeneracy:

$$\text{for all} \quad P \neq \mathcal{O} : \exists Q \in \mathbb{G}_2 \quad \text{such that} \quad \mathbf{e}(P, Q) \neq 1 \in \mathbb{G}_T$$
$$\text{for all} \quad Q \neq \mathcal{O} : \exists P \in \mathbb{G}_1 \quad \text{such that} \quad \mathbf{e}(P, Q) \neq 1 \in \mathbb{G}_T,$$

and *efficiency*, i.e., there exists an efficient algorithm $\mathcal{G}(1^\kappa)$ that outputs the bilinear group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, P, Q)$,

A pairing is said to be *symmetric* if $\mathbb{G}_1 = \mathbb{G}_2$, otherwise it is said to be *asymmetric*, i.e. $\mathbb{G}_1 \neq \mathbb{G}_2$. Moreover, Galbraith et al. [55] classify pairing instantiations into three basic types, (type-1) $\mathbb{G}_1 = \mathbb{G}_2$, (type-2) $\mathbb{G}_1 \neq \mathbb{G}_2$ where there exists an efficient isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$, and (type-3) $\mathbb{G}_1 \neq \mathbb{G}_2$ where an efficient isomorphism does not exist. Type-3 curves are the most efficient pairing friendly ones, so it is desirable for a scheme to work in such groups, i.e., $\mathbb{G}_1 \neq \mathbb{G}_2$ and the existence of an efficient isomorphism is not required.

Many pairings can be used to perform an algorithm $\mathcal{G}(1^\kappa)$, for example, Weil, Tate, Ate and Eta pairings are widely used due to their efficient computation. The choice of the pairing has significant performance impact as well as the choice of cryptographic library implementation, see Figure 2.4. The difference becomes more important with involving constrained devices such as smart cards, embedded devices, and smart meters, which are widely used in IoT, Industry 4.0 and smart metering systems.

There are several libraries with pairing-based cryptography support. Since we are interested about the best performance, and therefore, the fastest pairing calculation, we focused on libraries implemented in particular in C/C++ programming language. We installed selected libraries (Pairing Based Cryptography (PBC) [56], Multiprecision Integer and Rational Arithmetic Cryptographic Library (MIRACL) [57], University of Tsukuba Elliptic Curve and Pairing Library (TEPLA) [58], Efficient LIbrary for Cryptography (RELIC) [59] and MCL [60]) on embedded device

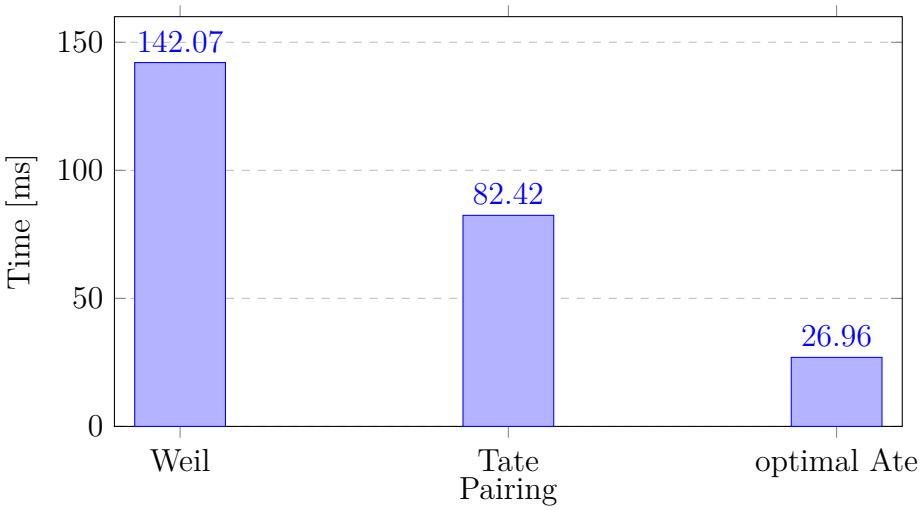

Fig. 2.4: The performance comparison of different pairings: Weil, Tate and optimal Ate, on ARMv8 processor (the Raspberry Pi 3, 32-bit OS) and using RELIC cryptographic library.

represented with ARM-based microcomputer (Raspberry Pi 3 with ARMv8 Cortex-A53 processor), and run the benchmarks using the 256-bit BN curve and 10-run means. The results are presented in Figure 2.5.

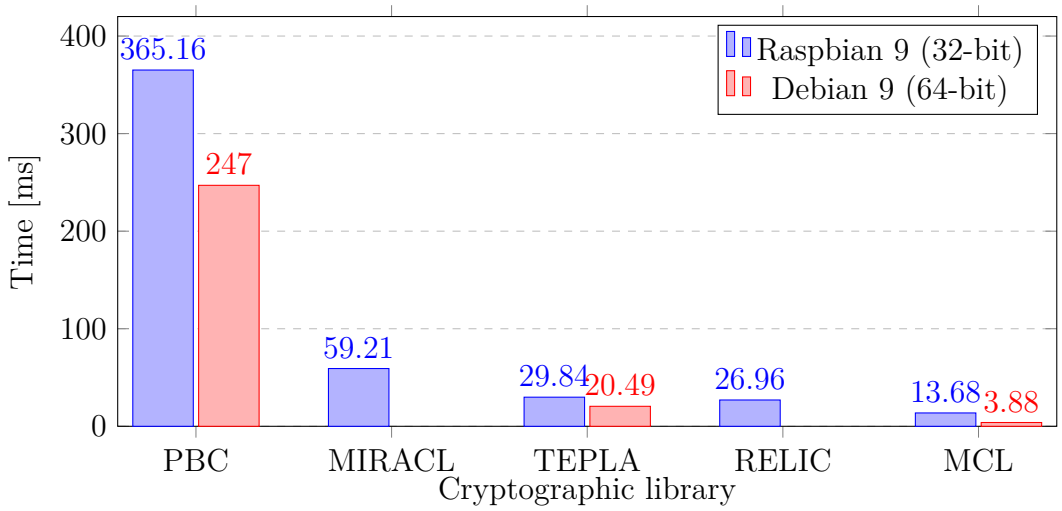

Fig. 2.5: The comparison of different cryptographic libraries from the point of view of bilinear pairing performance over BN 256-bit elliptic curve on the ARMv8 processor (the Raspberry Pi 3, 32-bit and 64-bit OS).

- The PBC [56] library is a free C library built on GNU Multi-Precision Arithmetic Library (GMP) that performs the underlying mathematical operations

of the pairing. The library provides routines such as elliptic curve generation, elliptic curve arithmetic and pairing computation. The library includes pre-generated pairing parameters with using different elliptic curves to achieve required properties of developed cryptosystems. In each case the curve group has a 160-bit group order, and corresponding *embedding degree k* of the curve.

- MIRACL Crypto Software Development Kit (SDK) is a free C library that is widely regarded by developers as the gold standard open source SDK for elliptic curve cryptography. Furthermore, MIRACL also enables developers to build security into highly constrained environments, including embedded, mobile applications and Supervisory Control And Data Acquisition (SCADA). Library has pre-generated different security levels of pairing friendly curves and implements type-1 and type-3 pairing. In case of type-3 pairing, where BN curves are included, the optimal Ate pairing is always used.
- TEPLA is a software library for development of applications or systems of cryptographic algorithms using pairings, developed by University of Tsukuba. TEPLA supports calculations on elements on Finite Fields (prime field of 254 bits, quadratic, 6th and 12th extension fields), calculations on elliptic curves (BN curves 254-bit) and calculation of pairings (Optimal Ate pairing on BN curves).
- RELIC is a modern cryptographic meta-toolkit written in C language. RELIC supports among others calculation on elliptic curves $E(\mathbb{F}_p)$ and $E(\mathbb{F}_{2^m})$ (NIST curves and pairing-friendly curves) and calculations of pairings and on related extension fields $\mathbb{F}_{p^k}$. The library includes Weil, Tate and Optimal Ate pairing, where Ate pairing is set as a default type.
- MCL is a library for pairing-based cryptography with support for x86-64 Windows, Linux and ARM/ARM64 Linux. The current version supports the Optimal Ate pairing over BN (254, 381, and 462-bit) curves and BLS12-381 curves.

In addition, the cryptographic designers and developers are equally interested in performance of related operations in each cyclic groups $\mathbb{G}_1, \mathbb{G}_2$ and/or $\mathbb{G}_T$. Therefore, we provide the performance benchmarks of these operations in Table 2.3, see [61] for more details.

2.4 Proof of Knowledge

The concept of proof of knowlege is frequently used in many privacy enhancing cryptographic schemes, such us group signatures, ring signatures and/or attribute-based credentials. The goal of this proof is for the prover $\mathcal{P}$ to convince the verifier $\mathcal{V}$

Tab. 2.3: Performance of group operations on ARMv8 processor (the Raspberry Pi 3, 32-bit and 64-bit OS). The required time of each operation is expressed in milliseconds.

	Raspbian 9 (32-bit)				**Debian 9 (64-bit)**			
	ecMul $\mathbb{G}_1$	**ecMul** $\mathbb{G}_2$	**mMul** $\mathbb{G}_T$	**mExp** $\mathbb{G}_T$	**ecMul** $\mathbb{G}_1$	**ecMul** $\mathbb{G}_2$	**mMul** $\mathbb{G}_T$	**mExp** $\mathbb{G}_T$
PBC	7.33	18.02	-	83.16	5.07	11.84	-	56.26
MIRACL	5.85	11.25	-	24.39	-	-	-	-
TEPLA	3.79	6.53	-	30.88	2.27	4.27	-	2.56
RELIC	3.10	9.07	-	17.90	-	-	-	-
MCL	3.30	6.17	-	8.98	0.74	1.67	-	2.63

Note: The **ecMul** denotes elliptic curve scalar multiplication, **mMul** is modular multiplication and **mExp** modular exponentiation operation.

about the veracity of the given statement. The statements about discrete logarithms in prime order groups can be easily proven using the Σ-protocols [62].

A simple, yet very often used protocol for proving the discrete logarithm knowledge is based on the Schnorr signature scheme [63]. Using this protocol, the prover proves his knowledge of a discrete logarithm with respect to public parameters $\mathbb{G}, g, q, c$. In particular, he proves the knowledge of $w : c = g^w \bmod p$, where p is the prime modulus, q is the group order and $\mathbb{G} = \langle g \rangle$ is a generator of $\mathbb{Z}_p^*$. The protocol is depicted in Figure 2.6.

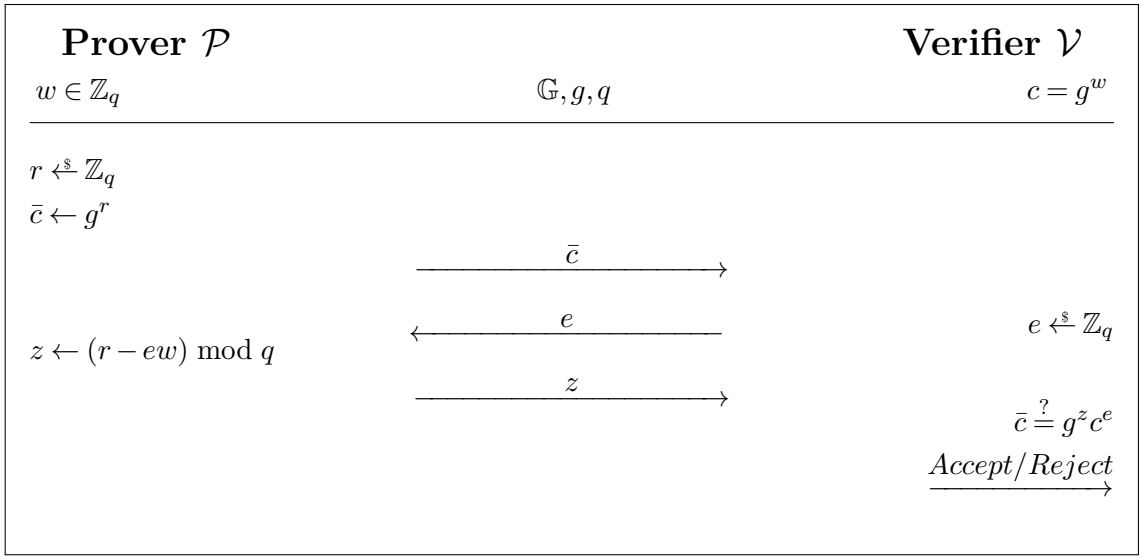

Fig. 2.6: Schnorr's proof of knowledge of discrete logarithm $\mathrm{PK}\{w : c = g^w\}$ in $\mathbb{Z}_p^*$.

The proof of the discrete logarithm knowledge is a simple 3-way protocol where the prover commits to a random number r in the first step, receives a challenge e in the second step and responds by z to the challenge in the third step. The protocol is Honest Verifier Zero-Knowledge (HVZK). Note that the verifier does not have to know the private input w of the prover to be able to verify its knowledge. We recall the properties of the protocol below.

Proof. Completeness: prover who knows w is always accepted: $\bar{c} = g^z c^e = g^{r-ew} g^{ew} = g^r g^{-ew} g^{ew} = g^r = \bar{c}$. □

Proof. Special Soundness: let's assume that cheating prover $\mathcal{P}^*$ is ready to answer at least 2 random challenges e, e' after committing to r without knowing w. Then, his responses z, z' must be accepted in verifier's checks:

$$\bar{c} = g^z c^e, \tag{2.20}$$

$$\bar{c} = g^{z'} c^{e'}, \tag{2.21}$$

we divide the equations 2.20 and 2.21, and get:

$$\frac{\bar{c}}{\bar{c}} = \frac{g^z c^e}{g^{z'} c^{e'}} \quad \Leftrightarrow \quad 1 = g^{z-z'} c^{e-e'}, \tag{2.22}$$

after multiplying both sides of equation 2.22 by $g^{-(z-z')}$ and raising to the power of $(e-e')^{-1}$, we get:

$$g^{(z'-z)(e-e')^{-1}} = c \tag{2.23}$$

and we get the discrete logarithm $w = (z'-z)(e-e')^{-1}$ that is easy to efficiently compute for the dishonest prover $\mathcal{P}^*$, thus we reached the contradiction because the cheating prover $\mathcal{P}^*$ unaware of w was assumed. □

Proof. Special Honest Verifier Zero-Knowledge: the ZK property is proven by proving the existence of the ZK simulator $M^*_{\mathcal{V}}$. The simulator $M^*_{\mathcal{V}}$ has the public input $\mathbb{G}, g, q, c$ and a challenge e. The output of the simulator is a transcript T of a protocol. A transcript for the protocol is of the form:

$$T = (\mathbb{G}, g, q, c)(\bar{c}, e, z).$$

The simulator $M^*_{\mathcal{V}}$ works in these following steps:

1. the simulator randomly chooses the response $z' \xleftarrow{\$} \mathbb{Z}_q$ on the challenge e.
2. the simulator computes the commitment $\bar{c}' \leftarrow g^{z'} c^e$.

The $M^*_{\mathcal{V}}$'s output $\bar{c}', e, z'$ is computationally indistinguishable from the real protocol output c, e, z. □

The protocol for proving the knowledge of a discrete logarithm described above can be extended to the discrete logarithm representation proof and discrete logarithm equivalence proof [17]. The principles and security proofs remain the same.

Interactive zero-knowledge proof is frequently used in authentication schemes, where a challenge e is generated by a verifier $\mathcal{V}$. On the other hand, *non-interactive* zero-knowledge proof is widely used in particular for signature scheme constructions. In this case, the challenge e is generated by the prover $\mathcal{P}$ with the use of secure hash function $\mathcal{H}$. To transform the interactive into a non-interactive zero-knowledge proof, the Fiat-Shamir heuristic [64] can be used. Non-interactive variant is more often called *Signature Proof of Knowledge* (SPK), because of included message in the proof, see Figure 2.7.

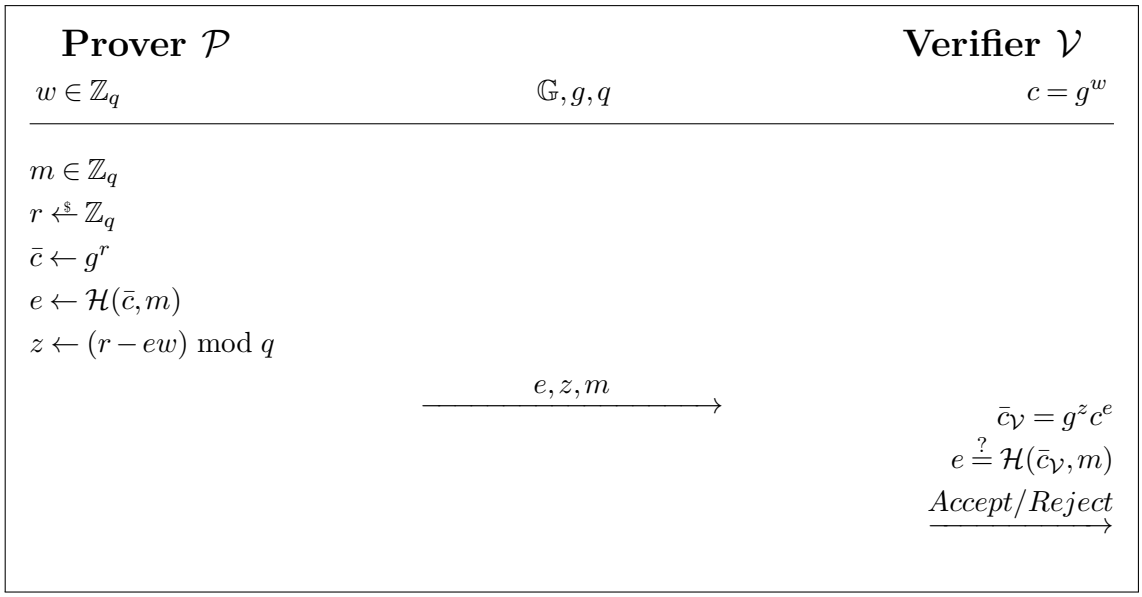

Fig. 2.7: Schnorr's signature proof of knowledge of discrete logarithm SPK$\{w : c = g^w\}(m)$ in $\mathbb{Z}_p^*$.

2.5 Weak Boneh-Boyen Signature

The weak Boneh-Boyen (wBB) signature scheme [25] can be used to efficiently sign (blocks of) messages. The signature scheme can be easily integrated with the zero-knowledge proofs so that the knowledge of signed messages (and signatures themselves) can be proven anonymously, unlinkably and utraceably. Furthermore, the wBB signatures were proven existentially unforgeable against a weak (non-adaptive) chosen message attack under the q-SDH assumption [65]. We recall the signing and verification algorithms below, the efficient proofs of knowledge are described, e.g., in [66].

`Setup`: On input security parameter κ, generate a bilinear group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2) \leftarrow \mathcal{G}(1^\kappa)$. Take $sk \xleftarrow{\$} \mathbb{Z}_q$, compute $pk = g_2^{sk}$, and output sk as the private key and $pk = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \mathbf{e}, pk)$ as the public key.

`Sign`: On input a message $m \in \mathbb{Z}_q$ and the secret key sk, output $\sigma = g_1^{\frac{1}{sk+m}}$.

`Verify`: On input the signature σ, the message m, and the public key pk, output 1 iff $\mathbf{e}(\sigma, pk) \cdot \mathbf{e}(\sigma^m, g_2) = \mathbf{e}(g_1, g_2)$ holds.

Showing the constant signature σ multiple times would make the authentication protocol linkable. All user sessions would be linkable to a single profile, which would make the resulting scheme very privacy unfriendly. To avoid linkability of signatures, users can only prove the knowledge of a valid signature by using the proof defined in [66]. In this proof, the user chooses a random value $r \xleftarrow{\$} \mathbb{Z}_q$ and computes randomized auxiliary values $\sigma' = \sigma^r$ and $\bar{\sigma} = {\sigma'}^{-m} g_1^r$. Then, the knowledge of a signature is proven by constructing the zero-knowledge proof $\pi = PK\{(m, r) : \bar{\sigma} = {\sigma'}^{-m} g_1^r\}$ and verifying $\mathbf{e}(\bar{\sigma}, g_2) = \mathbf{e}(\sigma', pk)$. The protocol is depicted in Figure 2.8. The verifier is convinced, that the user indeed knows a valid signature on a known message, although the proof does not release any of these values. That construction is perfect for privacy enhancing authentication schemes developing, because the users want to convince verifiers that they hold some cryptographic keys or/and personal attributes signed by registrars, in an anonymous, untraceable and unlinkable manner.

2.6 Okamoto-Uchiyama Encryption

Okamoto-Uchiyama (OU) public-key encryption scheme [67] can be used to encrypt (blocks of) messages. Moreover, the scheme is easy to integrate with zero-knowledge proofs so that committed value w in proof of knowledge $\mathrm{PK}\{w : c = g^w\}$ in $\mathbb{Z}_n^*$, where n is OU modulus, can be easily recovered. OU is based on the ability of computing discrete logarithms in a particular subgroup. In other words, the security is based on the assumption that the discrete logarithm problem is hard to compute in OU groups similarly as in RSA composite groups. However, if the factorization of the OU modulus $n = p^2 q$ is known, i.e., p, q large primes are known, the discrete logarithms can be efficiently computed and, therefore, it is possible to recover a *witness* w from a commitment $c = g^w \bmod n$. We recall the encryption and decryption algorithms below.

`Setup`: On input security parameter κ, generate two κ-bit primes p, q and compute the modulus $n = p^2 q$. According to current security standards [2], the modulus

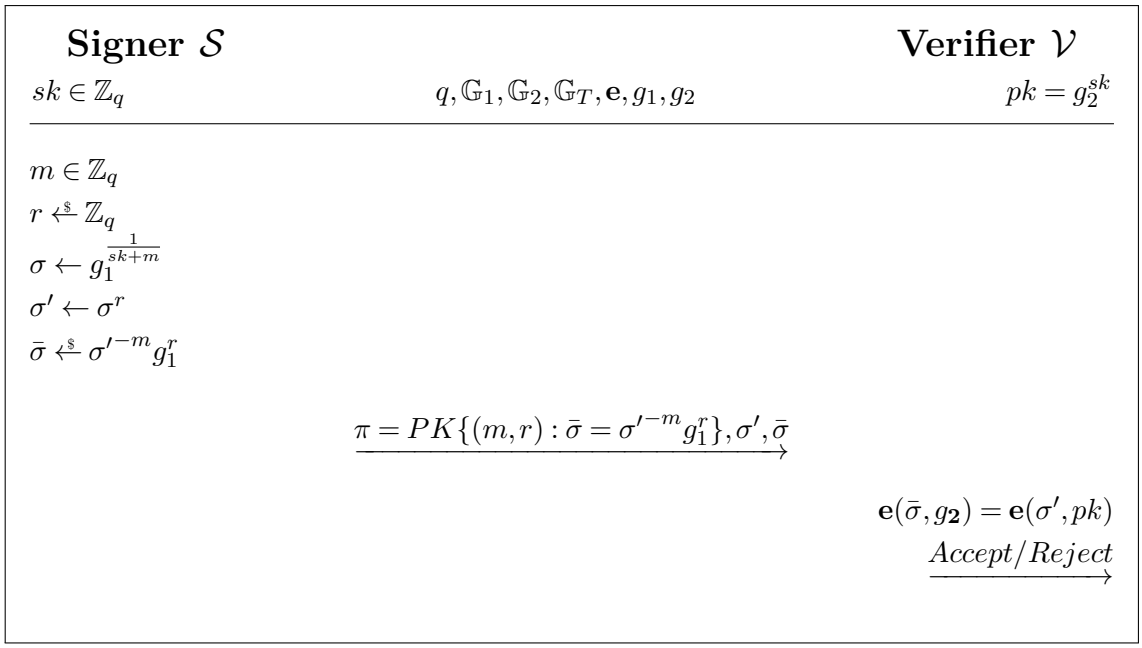

Fig. 2.8: Protocol for proof of knowledge of weak Boneh-Boyen signature $\mathrm{PK}\{(m,r) : \bar{\sigma} = \sigma'^{-m} g_1^r\}$ in $\mathbb{Z}_p^*$.

should be at least 2048-bit long, i.e. $|n| = 3\kappa = 2048$. Moreover, this algorithm randomly chooses the generator $g \xleftarrow{\$} \mathbb{Z}_n$, such that $g^{p-1} \not\equiv 1 \pmod{p^2}$ is of order p in $\mathbb{Z}_p^*$. Than, compute parameter $h = g^n \bmod n$. The pair (p,q) is securely stored as secret key sk, while the triplet (g,h,n) is published as the public key pk.

`Encrypt`: On input message $m \in \{0,1\}^{\kappa-1}$ and the public key pk, select $r \xleftarrow{\$} \mathbb{Z}_n$ and output ciphertext $c = g^m h^r \bmod n$.

`Decrypt`: On input the ciphertext c, and private key sk, output original message m as follows:

$$\begin{aligned} c' &= c^{p-1} = (g^m h^r)^{p-1} \\ &= (g^m g^{nr})^{p-1} = (g^{p-1})^m g^{p(p-1)rpq})^{p-1} = (g^{p-1})^m \bmod p^2 \end{aligned} \tag{2.24}$$

Based on Equation 2.24 it is straightforward that we can recover the message m only by solving DLP. Since, we know the factorization of n, i.e. primes p, q, where the value p is the trapdoor in the OU scheme, we can recover the message m from the following equation:

$$m = \mathrm{dlog}_g c = \frac{[(c^{p-1} \bmod p^2) - 1]/p}{[(g^{p-1} \bmod p^2) - 1]/p} \bmod p. \tag{2.25}$$

2.7 Algebraic MAC

Message Authentication Code (MAC) [16] is typically built from block ciphers and hash functions, which means that MACs have no algebraic structure and cannot be efficiently combined with zero knowledge proofs. Recently, Chase, Meiklejohn, and Zaverucha [68] introduced the notion of *algebraic* MAC, that relies on group operations instead of block ciphers and hash functions.

In terms of security, algebraic MACs are no different from traditional MACs. A MAC scheme consists of algorithms (`Setup`, `KeyGen`, `MAC`, `Verify`). `Setup` sets up the system parameters *par* that are given as implicit input to the other algorithms. `KeyGen` creates a new secret key, $\texttt{MAC}(sk, m)$ computes a MAC on message m, and `Verify` is used to verify MAC. We recall the security definitions due to Dodis et al. [69] and slightly strengthened by Chase et al. [68], and require completeness and Unforgeability under a Chosen Message and Verification Attack (UF-CMVA).

Definition 1. *A MAC scheme* $(\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{MAC}, \mathtt{Verify})$ *is complete if the following probability is negligible in* κ *for all messages* m*:*

$$\Pr\Big[\mathtt{Verify}(sk, m, \sigma) = 0 \mid par \xleftarrow{\$} \mathtt{Setup}(1^\kappa), \\ (ipar, sk) \xleftarrow{\$} \mathtt{KeyGen}(par), \sigma \xleftarrow{\$} \mathtt{MAC}(sk, m)\Big].$$

Definition 2. *A MAC scheme* $(\mathtt{Setup}, \mathtt{KeyGen}, \mathtt{MAC}, \mathtt{Verify})$ *is unforgeable under chosen message and verification attack if the following probability is negligible in* κ*:*

$$\Pr\Big[\mathtt{Verify}(sk, m^*, \sigma^*) = 1 \wedge m^* \notin Q \mid par \xleftarrow{\$} \mathtt{Setup}(1^\kappa), \\ (ipar, sk) \xleftarrow{\$} \mathtt{KeyGen}(par), (\sigma^*, m^*) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^{\mathtt{MAC}(\mathtt{sk},\cdot)}, \mathcal{O}^{\mathtt{Verify}(\mathtt{sk},\cdot,\cdot)}}(par, ipar)\Big].$$

3 State of the art

3.1 Group Signatures

A group signature is a cryptographic primitive widely used for providing user privacy and anonymity. The basic idea is to hide a user inside the bigger group of other users. Hence, a verifier is not able to learn any personal information (including the identity) of a signer. The only information that the verifier receives is whether the signer is a member of the group or he is not. In other words, an (anonymous) group signature allows users to sign a message on behalf of the group, in such a way that a signature does not disclose which user was signing the message. In the classical digital signature scheme, each signer holds his own keypair consisting of two specific keys: one private and one public key. The group signature scheme is similar to the classical digital signature scheme. In case of group signatures, there is one public key which is related with a set of private keys. A group signature scheme usually involves the following entities:

- **Users**: are group members that hold personal *group member private keys*. The users can sign the data anonymously on behalf of the group.
- **Verifiers**: are parties verifying the validity of the signatures by using the *group public key*.
- **Group Manager**: holds the *group manager private key*, adds new users into the group, and generates and issues the private keys to group members.
- **Revocation Manager**: holds the *group revocation private key*, revokes users from the system and discloses the user's identity. In some scenarios, the group manager can be merged with the revocation manager.

Group signatures have to provide classical *security properties* similarly to traditional digital signatures:

- **Authenticity**: ensures that the data was signed by a group member.
- **Integrity**: ensures that the signed data was not changed during data transfer between the User and the Verifier.
- **Completeness**: ensures that every valid signature generated by a group member is always verified correctly.
- **Soundness**: guaranties that every invalid signature always fails the verification process.
- **Unforgeability**: prevents the generation of a valid signature on a message without knowledge of the corresponding private key, e.g. using *chosen-message attacks*. Therefore, only a valid group member can generate a valid signature on behalf of the group.

- **Revocation**: allows the Revocation Manager to revoke a User from the system and thus prevents a revoked member to create valid signatures on behalf of the group.
- **Differentiation of group members**: ensures that each User has a different group member private key, and therefore he is uniquely identifiable by the Group Manager.

Furthermore, the signatures protect users' privacy, and hence should provide the following *privacy properties*:

- **Anonymity**: implies that Users sign data anonymously, hence a Verifier is not able to identify the signers.
- **Unlinkability**: guaranties that a Verifier, an eavesdropper nor group Users are not able to decide, whether two or more group signatures were generated by the same or different group members.
- **Coalition Resistance**: guarantees that no subgroup of group Users is able to generate a valid signature for non-member of the group, i.e. the signature is always openable to at least one member of the coalition.
- **Framing Resistance**: extends the Coalition Resistance property. The property guarantees that a subgroup of group Users cannot generate an openable signature to another group member even in co-operation with a malicious Group Manager.
- **Traceability**: allows the Group Manager to identify the signer of a given signature. Therefore, any valid group signature must be openable by Group (Revocation) Manager. Hence, the Group Manager can de-anonymize a User, link and trace signatures.
- **Unforgeable Traceability**: guaranties that Group (Revocation) Manager cannot falsely accuse a User of generating a signature on data he did not generate.

In addition to the *security* and *privacy* properties, the group signature may also define additional properties important for practical usage:

- **Dynamism**: users can be added to the group at any time as well as each member can be removed from the group without the need to issue new group member keys to remaining members. This procedure has no impact on the group public key size, nor on the size of generated signatures.
- **Efficiency**: most of the group signatures use demanding operations, such as modular multiplication and bilinear pairing operations. The main goal of the newest schemes is to avoid these operations in order to reduce required computation time to a minimum.

Currently, there are many group signature proposals that mostly fulfill security and privacy requirements described above. The first group signature schemes where introduced by Chaum and Heyst [70] in 1991. These signatures are important especially from the theoretical point of view, since they are very inefficient. The inefficiency is given particularly due to big sizes of signatures and public keys together with their linear dependence in the number of group members. Over time, newer schemes were proposed. These proposals focus not only on privacy requirements but also on efficiency and practical usage, i.e. dynamism, speed, size of signature and public key, their independence in the number of group members (system or black listed users), and revocation techniques. For more details see paper [71].

Group signatures became part of many current ICT applications and services where the protection of user privacy is required. Nonetheless, group signature schemes are usually even more computationally expensive and produce bigger signatures in comparison with standard digital signature schemes such as RSA, DSA or ECDSA. However, the signatures complexity is the key for their practical usage. The complexity becomes more crucial in current systems such as IoT, Vehicular Ad hoc Networks (VANET), Smart Grids, Smart Cities, and Industry 4.0. In each of these systems, group signatures can be beneficial for users who are concerned about their privacy. Moreover, these systems are usually formed by many constrained devices with power and memory restrictions which must be addressed in the newest proposals.

In fact, the area of group signatures is addressed by different international standards and research papers. For example, the German Federal Office for Information Security (BSI) [72] provides a comparison of 12 selected group signature schemes which comply the basic security and privacy requirements. The paper [73] compares the performance of two group signature schemes on mobile devices, namely pairing-based BBS [65] and non-pairing-based ACJT [74] group signature schemes. The results show that the signing and verification phases of both schemes take few seconds (up to 3 s) on smartphones with Android platform and 1 GHz CPU. In case of full pre-computation use, the signing phase is even faster (up to 50 ms), since it computes one hash function and few modular multiplications and additions. However, in the case of no pre-calculations, the verification of one signature takes 14.14 s for BBS and 1.4 s for ACJT. Another closely related work is the paper written by Potzmader et al. [75]. The authors investigate the performance of three anonymous digital signature schemes on mobile devices, that are all included in the ISO/IEC 20008-2:2013 standard [76]. This standard defines seven anonymous digital signature schemes in total and provides a general description of group public key mechanisms.

Based on the papers mentioned above and the current user's privacy requirements in many ICT applications, we provide comprehensive evaluation of group signature schemes and their practical usability in current ICT applications such as access control, data collection and data notification. In particular, we focus on smart phones implementations similarly to the paper [74]. Our results show the computational and space complexity of each scheme. This can serve as a indication of performance capability and complexity of the schemes on current smart cards. The contribution of this research was published in the paper [77].

Tab. 3.1: Evaluation of group signatures schemes.

Scheme	Sign Cost	Verify Cost	Sign Size	PK Size	Pairing	Assump.	Rev.
BBS [52]	$9E_{\mathbb{G}_1}+3E_{\mathbb{G}_T}$	$1\mathbf{e}+8E_{\mathbb{G}_1}+2E_{\mathbb{G}_2}+3E_{\mathbb{G}_T}$	$3\mathbb{G}_1+6\mathbb{Z}_p$ (1545 b)	$4\mathbb{G}_1+2\mathbb{G}_2$ (1050 b)	✓	q-SDH, DLIN, ECDL	*sk*
DP [78]	$8E_{\mathbb{G}_1}+3E_{\mathbb{G}_T}$	$1\mathbf{e}+7E_{\mathbb{G}_1}+2E_{\mathbb{G}_2}+3E_{\mathbb{G}_T}$	$4\mathbb{G}_1+5\mathbb{Z}_p$ (1559 b)	$4\mathbb{G}_1+2\mathbb{G}_2$ (1050 b)	✓	q-SDH, XDH, DLIN, ECDL	*sk*
HLCCN [79]	$7E_{\mathbb{G}_1}+5E_{\mathbb{G}_T}$	$1\mathbf{e}+5E_{\mathbb{G}_1}+2E_{\mathbb{G}_2}+4E_{\mathbb{G}_T}$	$3\mathbb{G}_1+5\mathbb{Z}_p$ (5600 b)	$6\mathbb{G}_1+2\mathbb{G}_2$ (1400 b)	✓	q-SDH, XDH, DLIN, ECDL	*sk*
ACJT [74]	$12E_{\mathbb{G}_n^*}$	$10E_{\mathbb{G}_n^*}$	$7\mathbb{G}_n^*+1\mathbb{Z}_q$ (7328 b)	$6\mathbb{G}_n^*$ (6144 b)	✗	SRSA, DDH, DL	*cred*, *rl*
CG [80]	$10E_{\mathbb{G}_n^*}$	$10E_{\mathbb{G}_n^*}$	$8\mathbb{G}_n^*+1\mathbb{Z}_q$ (8352 b)	$7\mathbb{G}_n^*+1\mathbb{Z}_q$ (7328 b)	✗	SRSA, DDH, DL	*cred*, *rl*
IMSTY [81]	$7E_{\mathbb{G}_n^*}+8E_{\mathbb{G}_1}$	$7E_{\mathbb{G}_n^*}+8E_{\mathbb{G}_1}$	$5\mathbb{G}_n^*+5\mathbb{Z}_p+1\mathbb{Z}_q$ (6155 b)	$7\mathbb{G}_n^*+4\mathbb{Z}_p$ (7848 b)	✗	SRSA, DH, ECDL	*cred*
HM GS [82]	$9E_{\mathbb{G}_n^*}$	$10E_{\mathbb{G}_n^*}$	$7\mathbb{G}_n^*+1\mathbb{Z}_q$ (7328 b)	$5\mathbb{G}_n^*$ (5120 b)	✗	DL, IF	*rl*

Note: $E_{\mathbb{G}_1}$ – EC scalar multiplication in $\mathbb{G}_1$, similarly $E_{\mathbb{G}_2}$ and $E_{\mathbb{G}_T}$, $\mathbf{e}$ – bilinear pairing, *sk* – group member private key, *cred* – credential, *rl* – revocation list.

Table 3.1 shows the summary of our group signature evaluation. The table shows the computational complexity of signature generation and verification. Furthermore, we provide signature and public key size comparison for equivalent security level of each scheme. We also depict which security assumptions are held by each scheme and how many pairings are computed. In the pairing-based schemes, $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, $\mathbb{Z}_p$ denote different groups with the following bitlengths: $|\mathbb{G}_1| = 175$ b, $|\mathbb{G}_2| = 175$ b and $|\mathbb{G}_T| = 1050$ b computed as $k \cdot |\mathbb{G}_1|$, where k is the embedding degree (e.g. $k = 6$). $|\mathbb{Z}_p| = 170$ b denotes the field size of an elliptic curve. In the non-pairing

schemes, $|\mathbb{G}_n^*| = 1024$ b denotes the multiplicative RSA group with exponents from $|\mathbb{Z}_q| = 160$ b. The revocation mechanism is the last considered parameter, since practical usability of the scheme depends on it as well. We denote user's private key update as sk, credential update as $cred$ and revocation list (black list) update as rl.

The implementation of the scheme is crucial, since it gives us a realistic view of scheme complexity on current computing devices (for example, PC, tablet, smart phone, and smart card). Our implementations were provided on different smart phone platforms and PCs, see [77] for more details. We employ two external cryptographic libraries: (1) Bouncy Castle [83] (cryptographic API and modular arithmetic operations) and (2) Java Pairing-Based Cryptography (JPBC) library [84] (pairing-based and elliptic curve operations).

The performance results on smart phone Nexus 5 LG are depicted in Figure 3.1. Note that non-pairing schemes show better performance results than paring-based schemes. Only in case of IMSTY scheme, we can see a significant increase of the verification and signing time. This is due to the use of elliptic curve operations in the scheme. In contrast to the theoretical assumptions, the elliptic curve implementations on current Android devices have significantly worse performance results.

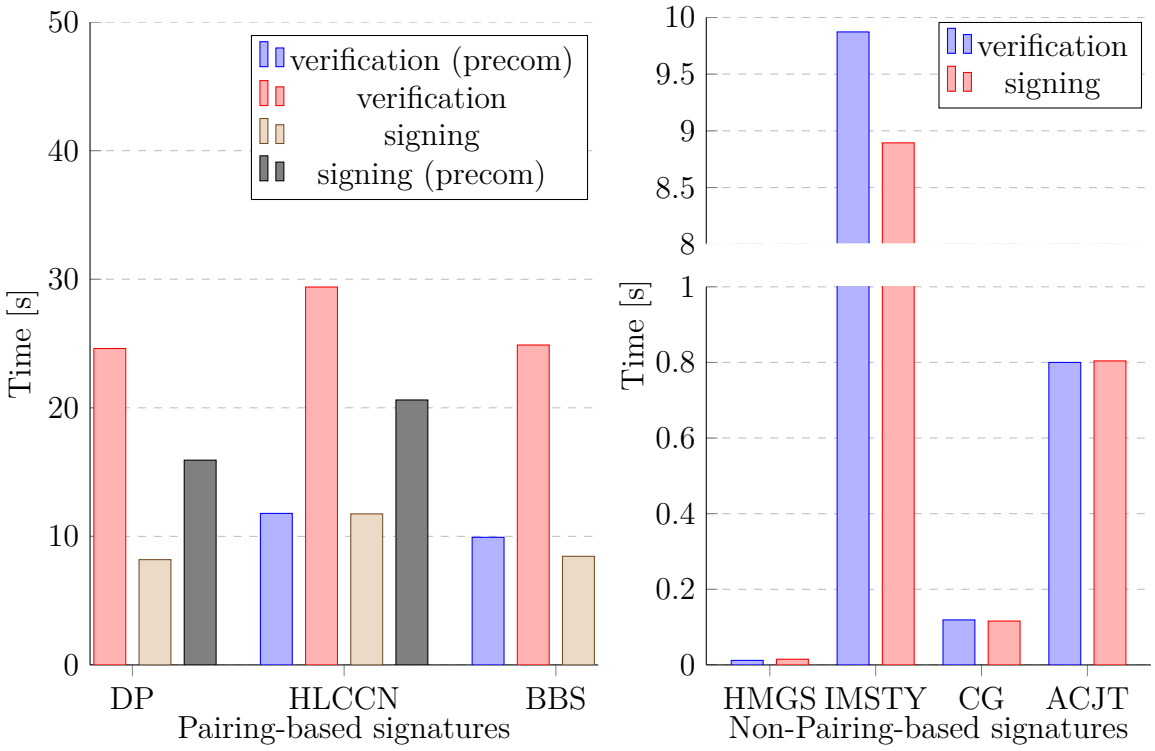

Fig. 3.1: Performance evaluation of current non-pairing-based and pairing-based group signature schemes on current smart phone (Nexus 5 LG).

3.2 Attribute-Based Credentials

Attribute-Based Credential is a cryptographic construction, that is a basic pillar of so-called attribute-based authentication schemes. In contrast to the classical authentication schemes based on identity, ABC schemes are more privacy-friendly, since they do not disclose user identity or other private information, that is not mandatory to gain an access to the required service. In many scenarios, it is not necessary to know a user identity to get an access. More important is to know, whether the user holds some personal attributes (his specific properties) which are directly related to the service scenario:

Public transport:	I have a `valid` ticket, and I applied for discount, since I am a `child`/`student`/`pensioner`.
Driving a car:	I have a `valid` driving licence of category B.
Access to university:	I have an access to my office and labs, since I am a `student`/`professor`.
Club membership:	I can make a padel court reservation and play, since I am member of the `Royal Tarraco` club, and I `paid` monthly fee.
Low emission zones:	I can enter to the London center, since I have a `diesel car` that meets the `Euro 6` emission standard.
Legal restrictions:	I can buy marijuana in Amsterdam, since I am older than `21` and I have `Dutch citizenship`.

Attributes are grouped together in a cryptographic (digital) *credential* as depicted in Figure 3.2. The credential is a cryptographic container for attributes signed by a trusted party. In general, we say, that credentials are issued and attributes are shown. Moreover, credentials usually include user's key, which provides non-transferability. In this context we can construct different credential types including set of common attributes:

Credential	**Attributes**
Identity:	Name, Surname, Social Insurance Number (SIN), Day of birth, Place of birth, Gender
Address:	Country, City, Street, Zip code
Student:	University, Faculty, Department, Personal ID
Health information:	Blood type, Allergy, Diagnosis

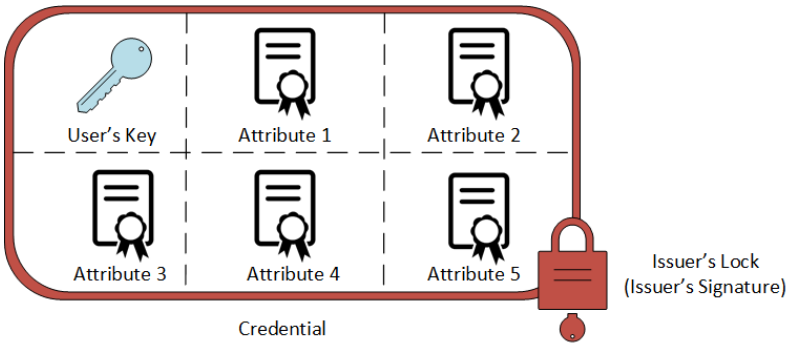

Fig. 3.2: Cryptographic credential construction.

Anonymous credentials hide the attributes, so, seeing a credential, no one can obtain any information about the attributes in it. Furthermore, the credentials allow user to authenticate himself without identification, and provide session unlinkability. Attribute-based authentication scheme normally involves following parties [85]:

- **Users**: are people equipped with a device that holds issued attributes. Attributes are issued by a trusted party (the Issuer), and their possession is anonymously proved to the Verifier.
- **Issuers**: are trusted authorities responsible for signing and providing credentials to Users.
- **Verifiers**: are parties verifying a possession of a subset of the available attributes on a device in order to authorize the transaction and provide an access to required service.
- **Manager**: is a trusted organisation that sets the rules for all involved parties. The organization is responsible for software and device management. In case of breaking rules, the Manager can revoke a (dishonest) User, or even disclose the (dishonest) User's identity.

Since the attribute based credentials are based on the credential-as-container concept, the following security properties should be hold [86]:

- **Authenticity**: ensures that a credential was issued by the Issuer and attributes belong to the User.
- **Integrity**: ensures that attributes included in the credential were not changed since they have been issued.
- **Confidentiality**: ensures attributes hiding, since a credential does not reveal the attributes that it contains.
- **Non-transferability**: prevents against credentials transferability among Users.
- **Revocability**: allows to the Third Trusted Party (TTP) to open the signature or the proof to disclose the User's identity, remove the User from the system, revoke User's credentials or revoke the unlinkability properties.

Furthermore, credentials protect Users' privacy and hence they should provide the following *privacy properties*:

- **Anonymity**: a User anonymously proves possession of attributes. Therefore, the User's identity and behaviour remain hidden.
- **Unlinkability**: each credential is fully randomized. This means that all credentials are mutually unlinkable even if the same credential is shown multiple times. Therefore, the property provides sessions unlinkability.
- **Untraceability**: guaranties that no information from the issue protocol can be used to link credentials when shown.
- **Selective disclosure of attributes**: allows a User to choose the attributes that he want to disclose. The rest of attributes remain hidden.

There is only several attribute-based credential schemes, e.g. U-Prove [87], Idemix [21] and Hajny-Malina [88]. The complete description of the schemes is beyond the scope of this theses, therefore we provide only brief protocols description including comparison of *security*, *privacy* and *performance* properties.

3.2.1 U-Prove

U-Prove is an anonymous attribute-based credential scheme [87] that belongs to Microsoft company. However, the scheme was first introduced and developed by Credentica company. The underlying cryptographic protocols were designed by Dr. Stefan Brands as a part of his Ph.D. thesis. Scheme security is based on discrete logarithm assumption. U-Prove uses same group as DSA signature scheme. In another words, U-Prove group is a prime order subgroup $\mathbb{Z}_q$ in the multiplicative group of a finite prime field $\mathbb{Z}_p^*$. The scheme uses a variant of the blind Schnorr signature [89] that is the key underlying cryptographic primitive of the scheme. Schnorr signature is used in an attribute issue protocol and guaranties untraceability of credentials by the Issuer. The attribute verify protocol uses the proof of knowledge protocols (cryptographic commitments and Σ-protocols), in particular a variant of the Schnorr protocol [90] is used. A U-prove user can selectively disclose a subset of his attributes, therefore a user is able to control how much information he releases. On the other hand, the scheme does not provide session unlinkability, since all credentials consist of a unique identificator Prover Information (PI) field. PI servers among others as a revocation handler, which allows to revoke dishonest users from the system. It is important to notice that the user real identity remains hidden and there is no way to disclose it. We provide a simplified description of the underlying protocols below. If more details are necessary, see the original paper [87].

`Setup`

$(pk_{\mathcal{I}}, sk_{\mathcal{I}}, par) \leftarrow$`Setup`$(1^{\kappa})$: the algorithm inputs the security parameter κ. It generates the cyclic group $\mathbb{G} = \langle g \rangle$ of prime order q satisfying $|q| = \kappa$. This algorithm also generates at random the Issuer's private key $sk_{\mathcal{I}} \xleftarrow{\$} \mathbb{Z}_q$ and computes $g_0 \leftarrow g^{sk_{\mathcal{I}}}$. Furthermore, for a set of possible attributes $\langle a_i \rangle_{i \in \mathcal{A}}$ it generates at random the generators $\langle g_i \rangle_{i \in \mathcal{A}}$ representing the part of the Issuer's public key $pk_{\mathcal{I}} = (g_0, \langle g_i \rangle_{i \in \mathcal{A}})$. It publishes the pair $(pk_{\mathcal{I}}, par)$ while $sk_{\mathcal{I}}$ is kept secret. This protocol is run by the Issuer.

`Issue_Att`

$(\sigma, h, s') \leftarrow$ `Issue_Att`$(pk_{\mathcal{I}}, sk_{\mathcal{I}}, par, \langle a_i \rangle_{i \in \mathcal{A}})$: the protocol is run by the User and the Issuer, as shown in Figure 3.3. The Issuer computes $h' = g_0 \cdot \Pi_{i \in \mathcal{A}} g_i^{a_i}$ as a aggregation of the attributes $\langle a_i \rangle_{i \in \mathcal{A}}$ and his public key $pk_{\mathcal{I}}$. He also generates the signature $z \leftarrow h'^{sk_{\mathcal{I}}}$. Moreover, the Issuer commits the blinded value $u \xleftarrow{\$} \mathbb{Z}_q$ to both generators $a \leftarrow g^u$, $b \leftarrow h'^u$. The User generates at random the values $s, v, \omega \xleftarrow{\$} \mathbb{Z}_q$, computes the secret key $s' \leftarrow s^{-1} \bmod q$, and gets U-Prove token $h \leftarrow h'^s$ and part of the signature $z' \leftarrow z^s$. Additionally, the User computes blind Schnorr signature commitment $c \leftarrow c' + \omega \bmod q$. At last the Issuer signs the commitment $r \leftarrow u + c \cdot sk_{\mathcal{I}} \bmod q$ which is the final part of the signature. The protocol outputs the U-Prove token h, the secret key s' and the signature $\sigma = (z', c', r')$.

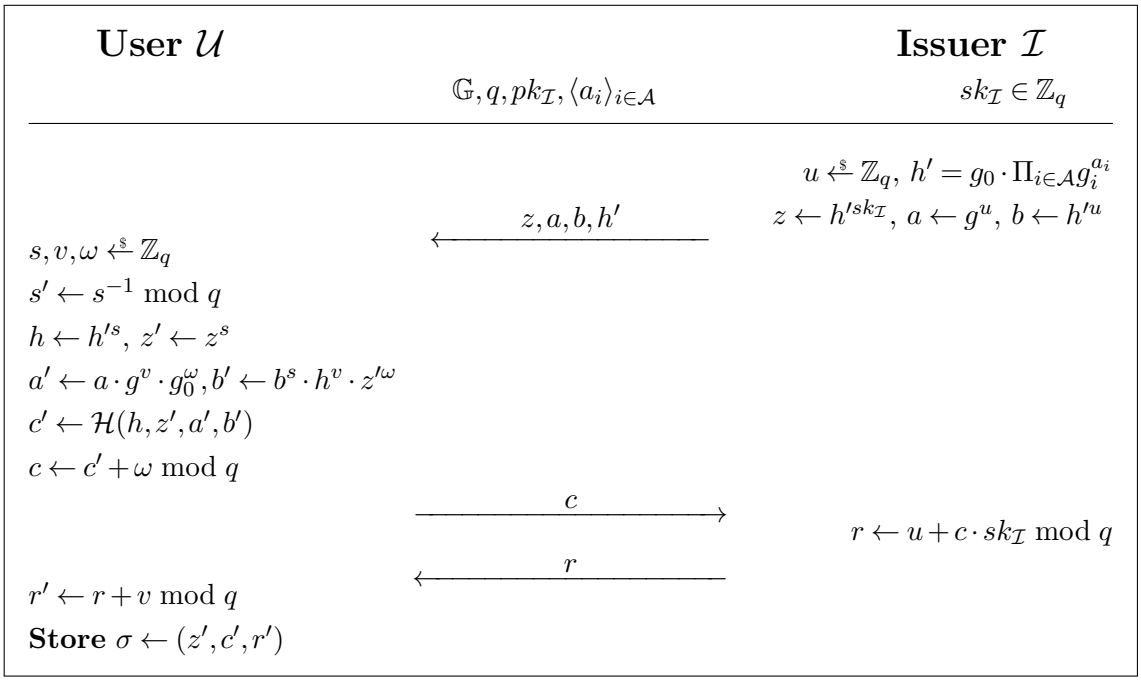

Fig. 3.3: U-Prove `Issue_Att` protocol

`Prove_Att`

$0/1 \leftarrow$`Prove_Att`$(\langle a_i \rangle_{i \in \mathcal{A}}, \sigma, h, s', par)$: the protocol is run by the User and the Verifier. The Verifier checks the correctness of the U-Prove token by restoring commitments $\hat{a}, \hat{b}$, and checking equality $c \stackrel{?}{=} \mathcal{H}(h, z', \hat{a}, \hat{b})$. Then, the Verifier checks the proof correctness. If the User knows the token key s' and all undisclosed attributes $\langle a_i \rangle_{i \notin \mathcal{D}}$, the Verifier accepts the proof, rejects otherwise. Therefore, this phase verifies, that the User knows the U-Prove token discrete logarithm representation $\text{PK}\{s', a_{i \notin \mathcal{D}} : h \leftarrow (g_0 \cdot \Pi_{i \in \mathcal{A}} g_i^{a_i})^s\}$. The protocol full notation is depicted in Figure 3.4.

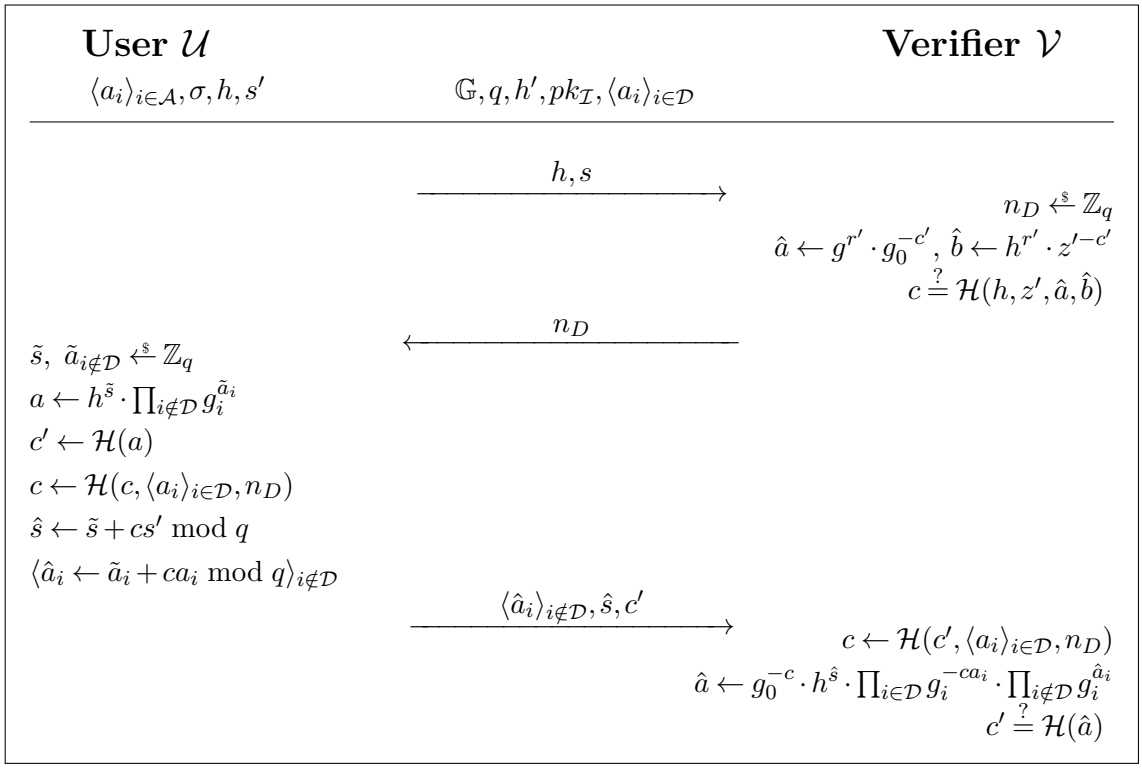

Fig. 3.4: U-Prove `Prove_Att` protocol.

`Implementation`

Currently there are only few implementations of the U-Prove protocol on smart cards. The most efficient implementation was provided on MultOS [91]. The attribute proving time depends on the number of stored attributes on the smart card and the number of disclosed attributes within the verification protocol, see Figure 3.5. However, in case of 5 attributes stored, the proving time is always under 1 s in each scenario.

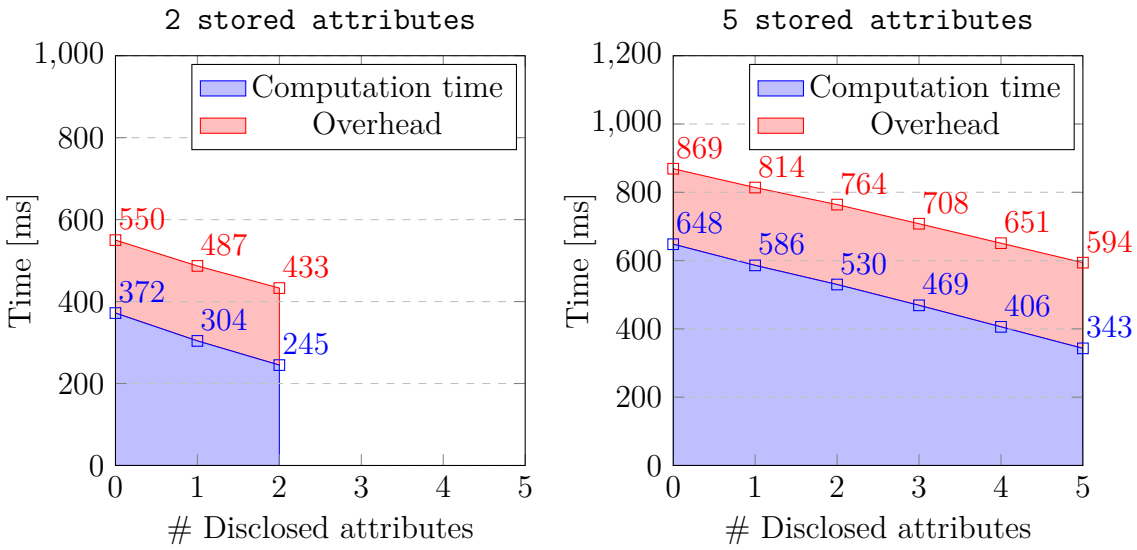

Fig. 3.5: U-Prove attributes proving time for different scenarios.

3.2.2 Idemix

Idemix (Identity Mixer) is an anonymous attribute-based credential scheme [21] developed by IBM Research Zurich. The scheme is based on Camenisch-Lysyanskaya signature [92] that allows the Issuer to sign User's attributes to construct a cryptographic credential within the issue protocol. The User randomizes and sends the credential to the Verifier and then anonymously proves possession of attributes to the Verifier by using zero-knowledge proof of knowledge protocols. The scheme security is held under *strong RSA assumption* in a cyclic group modulo composite $n = pq$, as well as in case of RSA cryptosystem. In contrast to U-Prove, the Idemix provides session unlinkability, that makes it impossible to track Users' movement and behaviour. Since every credential is randomized, there is no efficient revocation mechanism. Hence, the credentials may include time epoch information for limiting its validity or the scheme must be extended by external revocation scheme, e.g. [66].

`Setup`

$(sk_{\mathcal{I}}, par) \leftarrow$`Setup`$(1^{\kappa})$: the algorithm inputs the security parameter κ, see Appendix A for more details. It computes $n = pq$, where $p = 2p_{SG} + 1$, and $q = 2q_{SG} + 1$ are secure primes and p_{SG}, q_{SG} are Sophie-Germain primes. Then, this algorithm generates a cyclic group of quadratic residues $\mathbb{QR}_n = \langle S \rangle$ of order $p_{SG}q_{SG}$. It also computes bases $\langle R_i = S^{x_i} \rangle_{i \in \mathcal{A}}$ for each possible attribute $\langle x_i \rangle_{i \in \mathcal{A}}, x_z \xleftarrow{\$} [2, p_{SG}q_{SG} - 1]$ and an auxiliary value $Z = S^{x_z}$. The algorithm publishes $par = (n, S, Z, \langle R_i \rangle_{i \in \mathcal{A}})$ while $sk_{\mathcal{I}} = (p_{SG}, q_{SG})$ is kept secret. This protocol is run by the Issuer.

`Issue_Att`

$(\sigma, m_s) \leftarrow$ `Issue_Att`$(sk_{\mathcal{I}}, par, \langle a_i \rangle_{i \in \mathcal{A}})$: the protocol is run between the User and the Issuer, see Figure 3.6. At first, the User commits to his secret m_s in the commitment $U \leftarrow S^{v'} \cdot R_0^{m_s}$ and generates the corresponding proof of knowledge. Then, the Issuer checks the proof correctness. If the proof is valid, the Issuer aggregates required attributes with User's commitment to the credential $Q \leftarrow Z \cdot (U \cdot S^{v''} \cdot \prod_{i \in \mathcal{A}} R_i^{a_i})^{-1}$. The credential is signed $A \leftarrow Q^{1/e}$ and sent back to the User together with the credential public key e and the Issuer's blinded value v''. At last, the User computes the last peace of CL-signature $v \leftarrow v' + v''$. The protocol outputs the CL-signature $\sigma \leftarrow (A, e, v)$ and the credential key m_s as a User's private output.

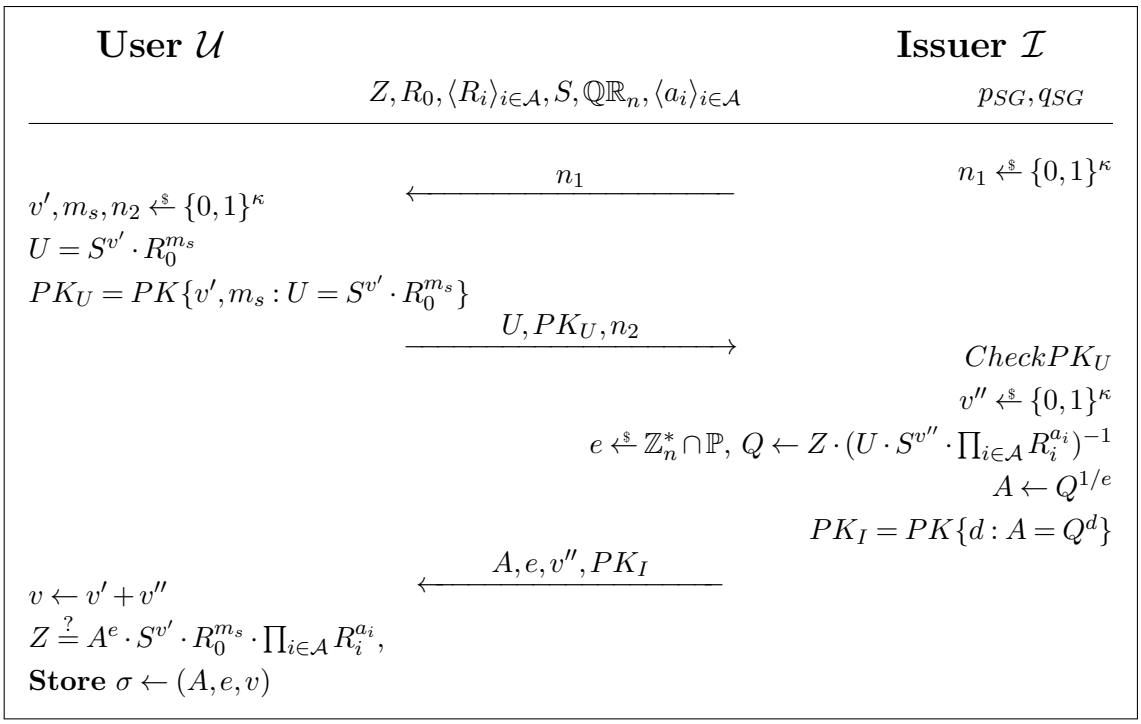

Fig. 3.6: Idemix `Issue_Att` protocol

`Prove_Att`

$0/1 \leftarrow$ `Prove_Att`$(\langle a_i \rangle_{i \in \mathcal{A}}, \sigma, m_s, par)$: the protocol runs between the User and the Verifier. The User sends a randomized CL-signature to avoid session linkability together with the proof of knowledge discrete logarithm representation of the CL-signature $PK\{e, v', a_{i \notin \mathcal{D}} : Z = A^e S^v R_0^{m_s} \cdot \Pi_{i \in \mathcal{A}} R_i^{a_i}\}$. The Verifier checks the proof correctness and therefore whether the User knows CL-signature over the attributes or not. The protocol full notation is depicted in Figure 3.7.

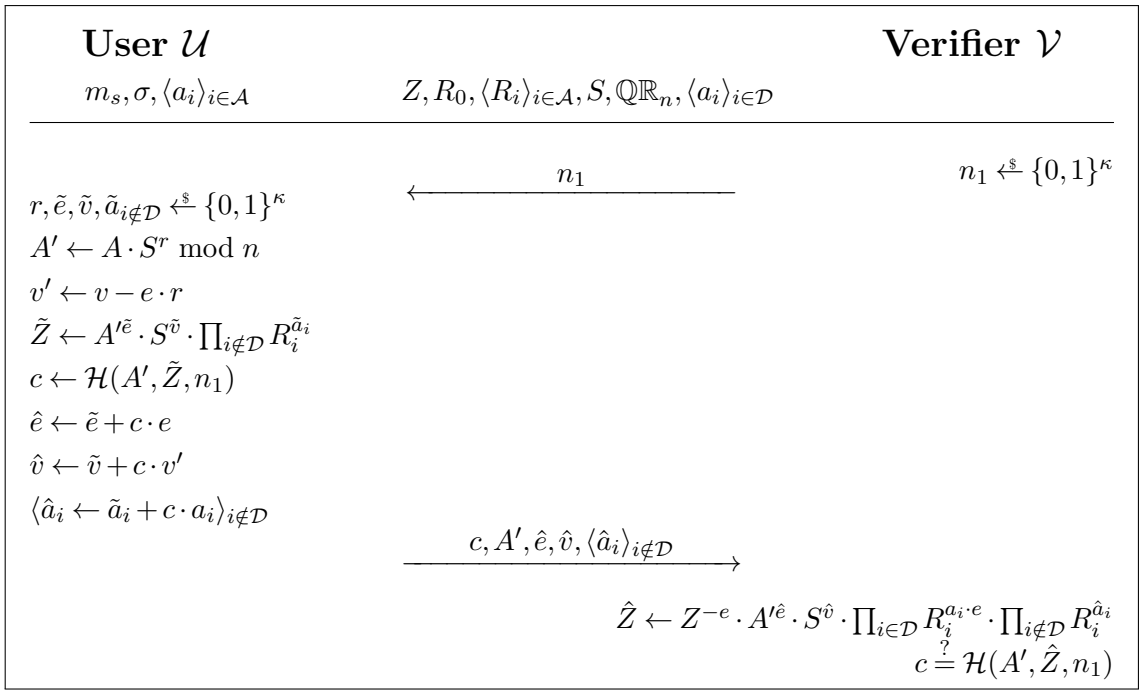

Fig. 3.7: Idemix `Prove_Att` protocol.

`Implementation`

Currently the most efficient implementation was provided on MultOS card [1], where the proof generation takes up to 1.5 s if 5 attributes were stored, see Figure 3.8.

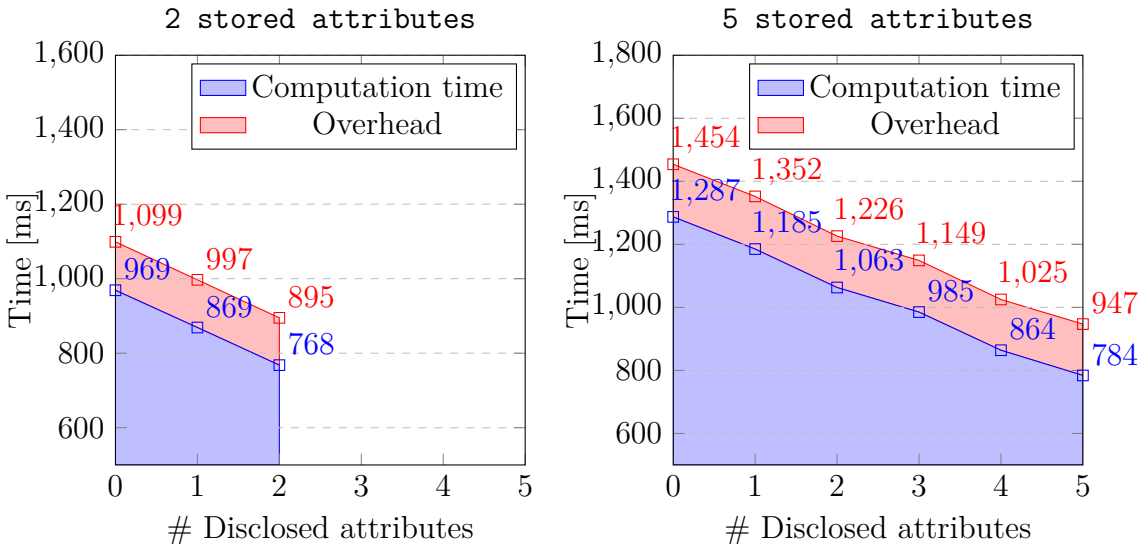

Fig. 3.8: Idemix attributes proving time for different scenarios.

3.2.3 HM12

Hajny-Malina (HM12) [88] is an attribute-based credential scheme with practical revocation developed by the Cryptology Research Group at Brno University of Technology in the Czech Republic. The scheme was first designed by Jan Hajny as a part of his Ph.D. thesis [93]. The scheme security is held under *discrete logarithm assumption* in Okamoto-Uchiyama group $\mathbb{OU}$, i.e. in a multiplicative cyclic group modulo composite number $\mathbb{Z}_n^*$, where $n = r^2 s$ and r, s are primes. The scheme uses the Okamoto-Uchiyama cryptosystem [67] as a key cryptographic primitive. This primitive allows the Manager to decrypt a proof generated by the User and thus disclose User's identity and revoke him from the system. For this reason, OU property acts mainly in `Issue_Att` and `Revoke` protocols, while the `Prove_Att` protocol runs fully over Σ-protocols (namely a Proof of Knowledge Discrete Logarithm Representation (PKDLR)). In contrast to previous schemes, the HM12 scheme provides practical revocation mechanisms, i.e. scheme itself allows to revoke issued credentials on the User's, Issuer's or Verifier initiatives. The scheme also supports revocation of credential unlinkability and User's anonymity. At the same time, there is required to involve more parties to the revocation process. For example, if Issuer, Manager and Verifier cooperate, they can revoke the User's anonymity while the cooperation only of Manager and Verifier allows to revoke session unlinkability and invalid credentials. The scheme is potentially weak against a cryptographic collusion attack, where more Users can in cooperation create a valid but unregistered User [94]. The weakness was solved in the protocol extension [95]. However, if we consider a tamper-resistance device (such as smart card), where the cryptographic keys are stored, we can avoid these collusion attacks.

`Setup`

$(par, K_{\mathcal{M}}, K_{\mathcal{I}}) \leftarrow$ `Setup`(1^κ): the algorithm inputs the security parameter κ, see Appendix B for more details, and generates the cyclic group $\mathbb{H}$ modulo big prime number p and the subgroup generators of order q, with $q|p-1$ similarly to DSA signature scheme. In addition, the Issuer generates the key pair $K_{\mathcal{I}}$ (consists of $sk_{\mathcal{I}}$ and $pk_{\mathcal{I}}$) for signing purposes. The Manager generates the cyclic group $\mathbb{OU}_n$ by specifying the modulus $n = r^2 s$, where r, s are secure primes (i.e. $r, s : r = 2r' + 1, s = 2s' + 1$ and r', s' are primes). Further, he gets randomly bases $g_1 \xleftarrow{\$} \mathbb{Z}_n^*$ of $ord(g_1 \bmod r^2) = r(r-1)$ in $\mathbb{Z}_{r^2}^*$ and $ord(g_1) = rr's'$ in $\mathbb{Z}_n^*$. The Manager chooses randomly his secrets $\langle S_i \rangle_{i \in \mathcal{A}}, S_2, S_3 \leftarrow \{0,1\}^\kappa$ such that $GCD(S_{i \in \mathcal{A}}, \phi(n)) = GCD(S_2, \phi(n)) = GCD(S_3, \phi(n)) = 1$ and computes attributes $\langle a_i = g_1^{S_i} \rangle_{i \in \mathcal{A}}$ and auxiliary values $g_2 = g_1^{S_2}$ and $g_3 = g_1^{S_3}$. The protocol outputs $par = (g_1, g_2, g_3, \mathbb{OU}_n, h_1, h_2, \mathbb{H})$ and $\langle a_i \rangle_{i \in \mathcal{A}}$ as public output, and values $(r, s, \langle S_i \rangle_{i \in \mathcal{A}}, S_2, S_3)$ as the Manager's private output.

`Issue_Att`

$(K_U) \leftarrow$ `Issue_Att`$(par, K_{\mathcal{M}}, K_I, \langle a_i \rangle_{i \in \mathcal{A}})$: the protocol is run between the User, the Issuer and the Manager, see Figure 3.9. In the first step, the User commits to his secrets w_1, w_2 to both generators $\bar{H} = h_1^{w_1} h_2^{w_2}$ and sends signed commitment with the proof of knowledge construction PK_{U1} to the Issuer. The Issuer checks the proof, signs the commitment and stores the copy. In the second step, the User commits to generators in $\mathbb{OU}_n$, i.e. computes $\bar{A} = g_1^{w_1} g_2^{w_2}$, and sends it to the Manager with signed value $\bar{H}$ from the Issuer and the proof of discrete logarithm equivalence PK_{U2}. The Manager checks the proof and issues the corresponding User (credential) partial key $\langle w_{3,i} \text{dlog}_{g_3}(a_i/\bar{A}) \rangle_{i \in \mathcal{A}}$ for each attribute by applying the OU trapdoor function. Finally, the Manager stores $\bar{A}, \bar{H}$ and $w_{3,i}$ to the secure database. The User combines partial secret keys and obtains User's (credential) key $K_U = \{w_1, w_2, \langle w_{3,i} \rangle_{i \in \mathcal{A}}\}$.

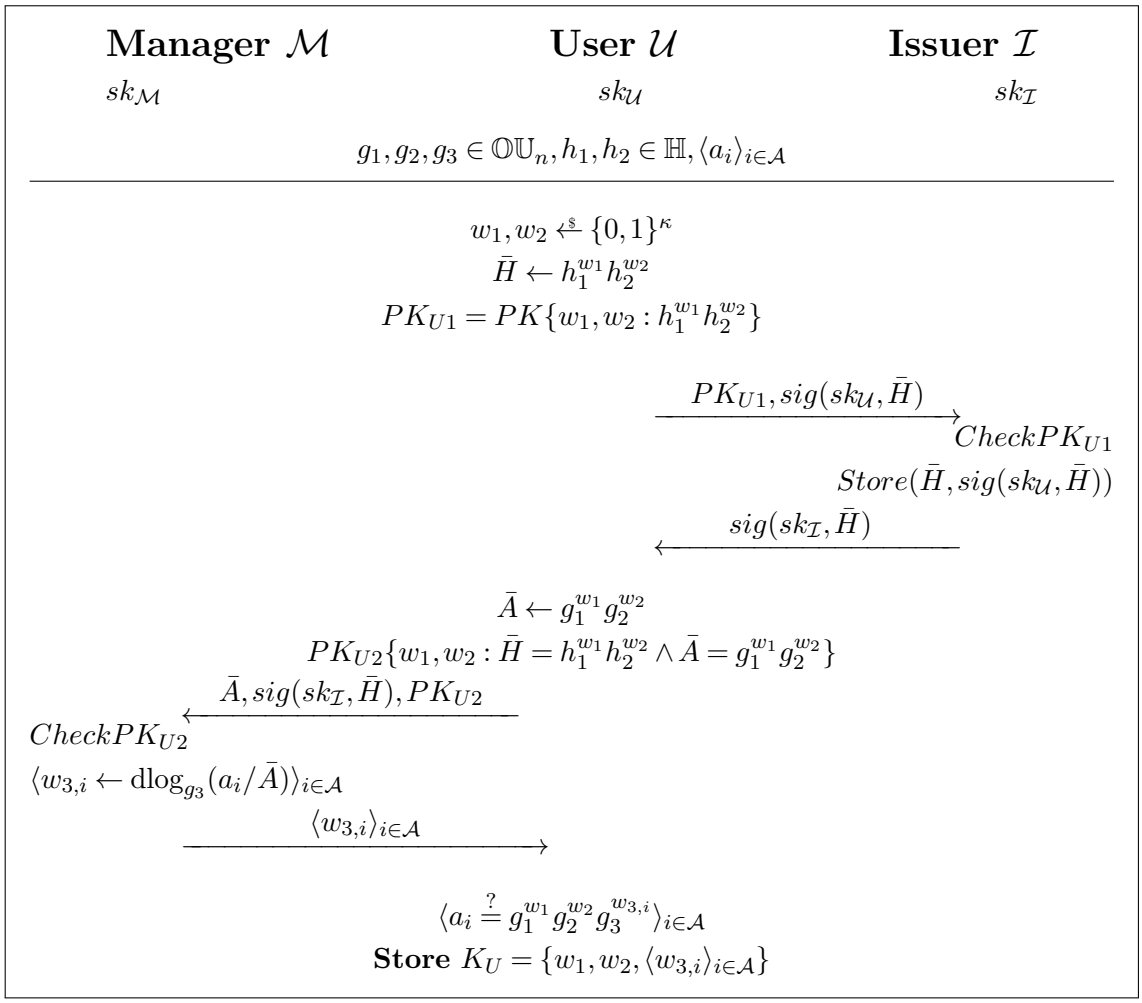

Fig. 3.9: HM12 `Issue_Att` protocol.

`Prove_Att`

$1/0 \leftarrow$ `Prove_Att`$(par, K_U, \langle a_i \rangle_{i \in \mathcal{D}})$: the protocol is run between the User and the Verifier, see Figure 3.10. The User proves the possession of required attributes. Hence, the User first discloses required attributes $\langle a_i \rangle_{i \in \mathcal{D}}$, and then proves the knowledge of the discrete logarithm representation of all of them, i.e. he proves the knowledge of the keys w_1, w_2 and all corresponding partial keys $\langle w_{3,i} \rangle_{i \in \mathcal{D}}$. For this reason, the User uses Σ-protocol constructions to generate the $proof$. The session unlinkability property is provided with a blinding value K_S changing in each session. The protocol provides revocation features, since the value K_S is committed in the commitment C_2 as well as partial keys $\langle w_{3,i} \rangle_{i \in \mathcal{D}}$ (revocation handlers) in the commitments $\langle C_{1,i} \rangle_{i \in \mathcal{D}}$. Commitments $\langle C_{1,i} \rangle_{i \in \mathcal{D}}$ and C_2 permit to check whether the User is blacklisted or not.

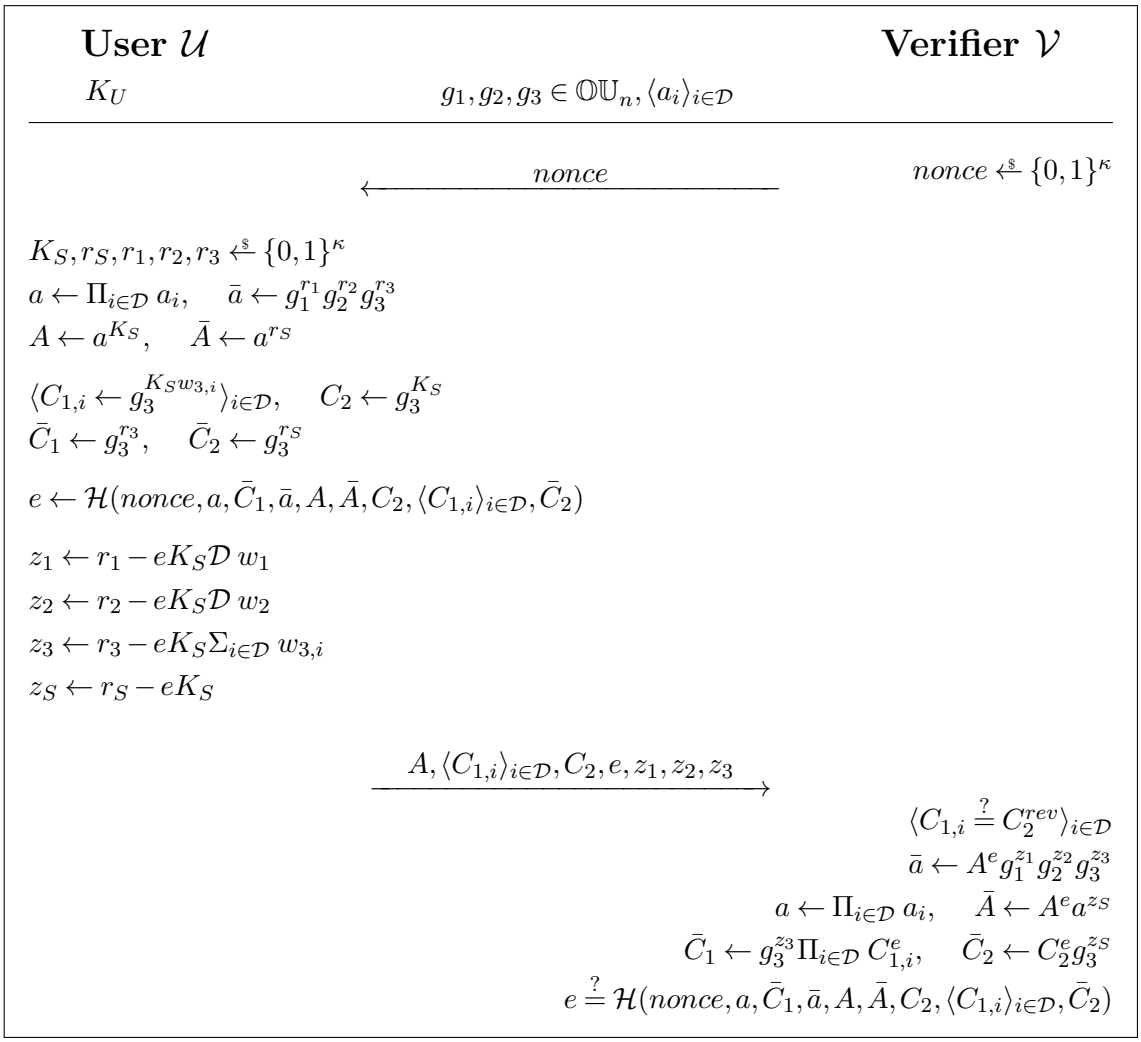

Fig. 3.10: HM12 `Prove_Att` protocol.

Revoke

$(rev) \leftarrow$ `Revoke`$(par, proof, K_{\mathcal{M}})$: the protocol is run between the Manager, the Issuer and the Verifier. The Verifier sends the commitments $\langle C_{1,i} \rangle_{i \in \mathcal{R}}$ ($\mathcal{R}$ denotes a set of attributes which must be revoked) and C_2 from the $proof$ to the Manager. The Manager restores partial keys as $\langle w_{3,i} = \text{dlog}_{g_3}(C_{1,i}/C_2) \rangle_{i \in \mathcal{R}}$. At last, the Verifier revokes the attributes by the publishing revocation record $rev = \langle w_{3,i} \rangle_{i \in \mathcal{R}}$ on a Black List. If necessary, the Manager sends corresponding commitment $\bar{H}$ to the Issuer, who can then revoke the User's anonymity.

Implementation

Currently the most effective implementation was provided on MultOS ML3 smart card [96] in a 1024-bit protocol variant. The verification time takes ca. 2.9 ms for one attribute disclosed. To provide comprehensive measurement of the scheme, we developed a smart card application that allows us to store and disclose up to 5 attributes. Our implementation (1024-bit version) was run on MultOS ML4 card. The time grows linearly with the number of disclosed attributes, see Figure 3.11. The number of stored attributes has no impact on the final time, since the attributes are not grouped in to the credential. Using the newer ML4 card instead of the older ML3 card, we reduced the attribute proving time by ca. 56%. However, the time complexity can be even more reduced by involving more computationally powerful devices. For example, the paper [97] uses 1392-bit protocol variant implementation on various smart phones to achieve the verification time under 100 ms.

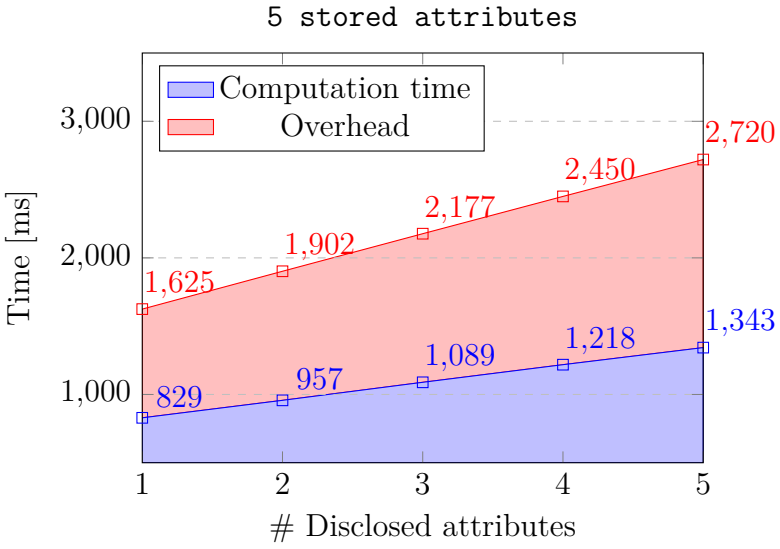

Fig. 3.11: HM12 attributes proving time for different scenarios.

Important to note: the time complexity grows linearly with the number of blacklisted attributes, since each attribute check involves one modular exponentiation.

3.3 Smart Cards

Smart cards evolved from simple memory cards to very efficient "*microcomputers*" with many applications. The security and portability of smart cards provide a fast way to ensure secure transactions, e.g. banking or e-business, and can be used in any system that requires secure authentication. In fact, smart cards are considered tamper-resistant storage devices protecting private keys and other sensitive information. Moreover, they contribute to the achievement of a safe environment for security-critical computation executions, as in the case of authentication, digital signature, and key exchange schemes. Since, our privacy-enhancing protocols, developed and described in this thesis, are primarily intended for card-based authentication and signature schemes, and they use elliptic curve constructions, we provide a short overview of the current state of the art of smart card technologies. We are mainly interested in hardware cryptographic support of elliptic curves, and related modular arithmetic and cryptographic functions on current smart cards. We expect that the results of this section may serve cryptographic protocol designers to get better overview on how fast privacy-enhancing protocols can be executed on the current smart cards. The main contribution in this section has been published in articles [12] and [98].

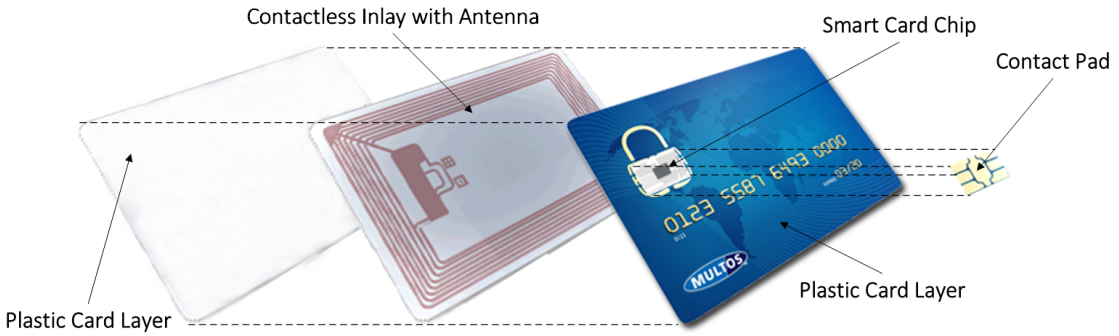

Fig. 3.12: Smart card construction.

A smart card is a small plastic card, typically of credit-card size, that has an embedded integrated circuit, see Figure 3.12. The circuit can store and/or process data (microprocessor and/or memory chip is used) and communicate with a terminal via communication interface (i.e. antenna or contact pad). Based on the embedded circuit, we distinguish two broad smart card categories:

- **Memory Cards**: can only store data, since they have no processor on the card for data processing. These cards are suitable for systems where they perform only fixed operations.
- **Microprocessor Cards**: are equipped with a 8-bit, 16-bit or 32-bit processor, therefore, they can also process data. These cards are also often called "*chip cards*" or "*smart cards*" and they can be classified as follows:
  - **Cryptographic Cards**: have also an embedded cryptographic processor "*co-processor*" in order to accelerate some cryptographic algorithms on the card, such as Advanced Encryption Standard (AES), RSA, ECDSA, and ECDH.
  - **Programmable Cards**: allow developers to install and run their own applications on the card. The most widespread programmable smart card platforms are Java Card, MultOS, Basic Card, and .NET Card.

The smart card interface defines the card usability. Generally, we classify smart cards into two broad classes; *contact smart cards* in accordance with the international standard ISO/IEC 7816, and *contactless smart cards* in accordance with the international standard ISO/IEC 14443. Regardless of the platform operation system, all smart cards communicate with a terminal via Application Protocol Data Units (APDU) according to ISO/IEC 7816-4. The communication protocol between a smart card and a terminal follows the client-server model, where a card acts as a server and a terminal acts as a client, i.e. the card receives an *APDU command* and replies to the terminal with an *APDU response* message. The message has a byte array representation, with a maximal payload size given by the transmission protocol used, i.e. `T=0` (255 bytes), or `T=1` (65 535 bytes) according to the ISO/IEC 7816-3. The smart card memory chip usually consists of three types of memory: the Electrically Erasable Programmable Read-Only Memory (EEPROM) for applets storage (*tens* of KB), the Random Access Memory (RAM) to store temporary (session) data (*units* of KB), and the Read-Only Memory (ROM) to store the smart card operating system (*hundreds* of KB). RAM memory is usually faster and more secure against a power analysis attacks.

3.3.1 Application Programming Interface

Smart cards are a closed platform, i.e., it is not usually possible to upgrade cryptographic libraries on the card. Cryptographic support differs according to the smart card platform: Java Card, MultOS, Basic Card, .NET Card, version of the operating system and the smart card implementation itself. In fact, there is often an inconsistency between the platform specification and the real implementation

of smart cards' Application Programming Interface (API) due to the implementer company, e.g. NXP, Gemalto, Giesecke & Devrient, Feitan, Oberthur, Ubivelox, Hitachi, Samsung, MultOS International, ZeitControl GmbH.

Table 3.2 shows the support of cryptographic functions on different smart card platforms. These types of security functions are: symmetric cryptography (`SymmetricCrypto`), asymmetric cryptography (`AsymmetricCrypto`), hash functions (`MessageDigest`), random number generator functions (`RandomData`), modular arithmetic operations (`ModularArithmetic`) and elliptic curve operations (`EllipticCurve`). The table presents the basic overview of supported functions, since the platforms usually offer various operating system versions and smart card implementations.

Advanced cryptographic protocols usually require modular arithmetic operations such as *multiplication* and *exponentiation* with big integers, as well as operations over elliptic curves, including *point addition* and *scalar multiplication*. These operations are provided by MultOS and Basic Card platforms. Java Card offers many standard cryptographic schemes, but the underlying mathematical operations, such as modular arithmetic and elliptic curve operations, are still missing.

Since we are mostly interested in elliptic curve cryptography and related underlying mathematics operations, we provide a brief description of the most popular smart card platforms and their elliptic curve cryptography algorithms support.

Java Card

Java Card (JC) technology developed by Oracle corporation defines a multi-application smart card platform where each applet (run application) is written in JC language, which is a cut-down version of the Java programming language. JC technology is currently one of the most widespread smart card technologies that is easy to implement, since various development environments, e.g. NetBeans, Eclipse or JCSSuite (Giesecke & Devrient), can be used to write, debug and install JC applets on the card. Same JC applet can run on different smart cards due to Java Card Virtual Machine (JCVM) and JC API, which provide hardware abstraction layer between applet and smart card implementation. All variables are stored in the EEPROM memory by default, however, JC also allows developer's access to the RAM memory through Java Card API to create transient data, e.g. `JCSystem::makeTransientByteArray()`.

JC framework supports a large number of different cryptographic algorithms, including ECC algorithms. ECC is available from version 2.2. However, only elliptic curves of the Short Weierstrass form over prime field $\mathbb{F}_p$ and the Koblitz form over binary field $\mathbb{F}_{2^m}$, both in affine coordinates only, are available. These curves can be used only within the supported protocols, i.e. ECDH and ECDSA. JC does not

Tab. 3.2: Cryptographic and mathematical support of smart card platforms.

	Java Card	**Basic Card**	**MultOS**	**.NET Card**
`Symetric Crypto`	DES, TDES, AES (keys up to 256 b), SEED, CBC/ECB modes, CMAC, HMAC	DES, TDES, AES (keys up to 256 b), CBC/CFB/OFB/EAX modes, OMAC	DES, TDES, AES (keys up to 256 b), SEED, CBC/ECB modes	DES, TDES, AES (keys up to 256 b), ECB/CBC modes
`Asymetric Crypto`	RSA (up to 4096 b), DSA (up to 1024 b), ECDH, ECDSA (up to 512 b)	RSA (up to 4096 b), ECDSA, ECDH, ECNR signature (up to 521 b)	RSA (up to 2048 b), ECDH, ECDSA, ECIES (up to 512 b)	RSA (up to 2048 b)
`Message Digest`	MD5, RIPEND160, SHA-1, SHA-2, SHA-3 (JC 3.0.5)	SHA-1, SHA-2 (up to 512 b)	SHA-1, SHA-2 (up to 256 b)	MD5, SHA-1, SHA-2 (up to 256 b)
`Random Data`	Pseudo RND, TRNG (JC 3.0.5)	4B RND function, TRNG	TRNG	Pseudo RNG, TRNG
`Modular Arithmetic`	**not supported**, exponentiation (RSA encryption), multiplication (only software solution with RSA tunnel: $ab = [(a+b)^2 - a^2 - b^2]/2)$	supported (up to 16 kB)	supported (up to 2048 b)	**not supported**, same as **Java Card**
`Elliptic Curve`	**not supported**, JC 3.0.5: scalar multiplication (`ECDH_PLAIN_XY`), point addition (`PACE_GM`)	supported (up to $\mathbb{F}_{521}$ or $\mathbb{F}_{2^{193}}$)	supported (up to $\mathbb{F}_{512}$)	**not supported**

support other algorithms over EC and there is no direct access to the underlying arithmetic operations. Version 3.0.1 allows to use bigger curves $E(\mathbb{F}_p)$ compared to the previous versions (up to 384 bits instead of up to 256 bits), and also permits to obtain the x-coordinate of the curve point $P(x, y)$, which is computed by the key agreement algorithm in the ECDH scheme. Moreover, the version supports SHA-2 hash algorithm within ECDSA signature scheme.

Version 3.0.4 supports even bigger curves $E(\mathbb{F}_p)$, i.e. up to 521 bits, and from version 3.0.5, the cryptographic algorithms `ALG_EC_PACE_GM` and `ALG_EC_SVDP_DH(C)_PLAIN_XY` are available. These algorithms bring a great advantage since they can be (ab)used to compute point addition and point scalar multiplication without cards explicit support. `ALG_EC_PACE_GM` algorithm can be used to compute point addition, i.e. given two points of a supported elliptic curve $A = (x_A, y_A)$

and $B = (x_B, y_B)$, it is possible to compute $C = (x_C, y_C) = (x_A, y_A) + (x_B, y_B)$. `ALG_EC_SVDP_DH(C)_PLAIN_XY` algorithm can compute scalar multiplication (point multiplication) $C = (x_C, y_C) = a \cdot (x_B, y_B)$, where $a \in \mathbb{F}_q$. JCs can also include extending API from the manufacturer, like in the case of Next eXPerience (NXP) Semiconductors cards. These cards, in particular cards with Operating System (OS) NXP Java Card OpenPlatform (JCOP) v2.4.1 and newer, contain the `com.nxp.id.jcopx` package which implements special classes such as `KeyAgreementX` that allows to use `ALG_EC_SVDP_DH(C)_PLAIN_XY` algorithm for scalar multiplications and `ECPoint` that computes the point addition over an elliptic curve. Unfortunately, no JC framework with 3.0.5 version has been created yet and the developers tools for using NXP API are missing.

MultOS

MultOS is the multi-application smart card operating system developed by MultOS Consortium. The applets are written in plain C language, however, MULTOS Executable Language (MEL) assembly can be used as well. The main development tools for writing, debugging and installing applets are MultOS Utility (MUtil) and MultOS SmartDeck both distributed by MultOS Consortium. In addition to MultOS International, there are only few more implementers of MultOS smart cards: DNP with Hitachi, SAMSUNG SDS and Ubivelox. The implementers develop their own MultOS cards supporting specific cryptographic APIs according to their own applications. Similarly to Java Card, same MultOS applet can run on different smart cards due of virtual machine called Application Abstract Machine (AAM). MultOS applets are divided into three distinct memory types: `melpublic` serves as input/output buffer for applications (APDU exchange), `melsession` stores temporary data (session data), which are both placed in RAM memory, and `melstatic` placed in EEPROM memory, where the application code and static data are resided. It is important to note that the session data size is fixed and must be declare before installing any application on the card.

The ECC support is available from MultOS version 4.2. MultOS supports only elliptic curves over prime field $\mathbb{F}_p$ (up to 512 bits). In particular, MultOS supports only elliptic curves of Short Weierstrass form with points represented in affine and projective coordinates. From version 4.2 to version 4.5.1 (last implemented), the following basic operations over EC are available: point addition, inverse, multiplication, verify point, convert representation (affine, projective) and equality test of two points. Moreover, MultOS supports also cryptographic algorithms such as ECIES, ECDH, ECDSA and key pair generation. On the other hand, all the point operations are optional in MultOS and the implementation is not mandatory, for example

MultOS International implements cards "*ML3 Generic family*" with basic support of complex ECC protocols such as ECDH, EC key pair generation and ECDSA, but the implementation of underlying arithmetic operations is completely missing, see Table 3.5 for more details.

Basic Card

Basic Cards distributed by ZeitControl (ZC) are programmable smart cards with each applet written in ZC-Basic language. Basic Card Development Environment distributed by ZeitControl is used as a development tool. Currently, the ZC5, ZC6, ZC7, ZC8-series of Basic smart cards are available. Similarly to previous smart card platforms, developers may choose the memory type: `Public`, `Static` data which are both placed in volatile RAM memory, and `Eeprom` data which are placed in permanent EEPROM memory.

The ECC support is available from the ZC5-series, where only complex ECDH and ECDSA algorithms over the binary fields $\mathbb{F}_{2^{167}}$ and $\mathbb{F}_{2^{211}}$ are supported. From the ZC7-series, the support of elliptic curves over prime field $\mathbb{F}_p$ is available, together with related underlayer mathematical operations, such as point addition `ECpAddPoints` and scalar multiplication `ECpMultiplyPoint` on $E(\mathbb{F}_p)$. Both ZC7-series (Professional cards) and ZC8-series (Multi Applications cards) support $E(\mathbb{F}_p)$ for p up to 544 bits. These cards implement all the (fourteen) curves recommended by Brainpool Standard [36] (short Weierstrass form and twisted curves) and all the (five) prime curves recommended by NIST (FIPS [22]). Moreover, these cards hold precomputed curve points related to the curve based point that allows to accelerate operations with the based point (in particular scalar point multiplication). Other curves are not supported by default, however it is possible to pay an additional fee to ZeitControl and get a support of required elliptic curves.

.NET Card

.NET Cards are multi-application smart cards with applications written in C# language. These cards use the Gemalto .NET Smart Card Framework, which is cut-down version of .NET Framework. The .NET applets are executed within .NET virtual machine, which ensures the portability of applications between different smart cards. For developing and installing applets on .NET Cards, the Microsoft Visual Studio (2008-2010) with pre-installed special smart card plug-in can be used, i.e. Gemalto .NET SDK v2.2 (Card Explorer Tool). The main implementer of the cards is the Gemalto company. These cards contain name-space `System.Security.Cryptography` which includes different cryptographic classes such

as RSA algorithm up to 2048 bits, DES, 3DES and AES ciphers and hash algorithms, e.g. MD5, SHA-1 and SHA-2. However, the support of ECC is completely missing.

In conclusion, JCs and Basic Cards support short Weierstrass and Koblitz forms. Additionally, MultOS supports also short Weierstrass form on *affine* $\mathcal{A}$ or *projective representation* $\mathcal{P}$ of the EC point. An overview of elliptic curve cryptography support based on smart card platform is depicted in Table 3.3.

Tab. 3.3: Elliptic curve cryptography support on smart card platforms.

	Version	`ecAdd`	`ecMul`	`ecInv`	ECIES	ECDH	ECDSA	$\lvert\mathbb{F}_p\rvert/\lvert\mathbb{F}_{2^m}\rvert$	Space
Java Card	JC 2.2.2	✗	✗	✗	✗	✓	✓	192/193	$\mathcal{A}$
	JC 3.0.1	✗	✗	✗	✗	✓	✓	384/193	$\mathcal{A}$
	JC 3.0.4	✗	✗	✗	✗	✓	✓	521/193	$\mathcal{A}$
	JC 3.0.5	✓!	✓!	✗	✗	✓	✓	521/193	$\mathcal{A}$
	JCOP2.4.1	✓!	✓!	✗	✗	✓	✓	320/–	$\mathcal{A}$
MultOS	4.2	✓	✓	✓	✓	✓	✓	384/–	$\mathcal{A}$,$\mathcal{P}$
	4.3.1 - 4.5.1	✓	✓	✓	✓	✓	✓	512/–	$\mathcal{A}$,$\mathcal{P}$
BasicCard	ZC5, ZC6	✗	✗	✗	✗	✓	✓	–/211	$\mathcal{A}$,$\mathcal{T}$
	ZC7, ZC8	✓	✓	✗	✗	✓	✓	544/211	$\mathcal{A}$,$\mathcal{T}$
.NET	Gemalto .NET 2.0	✗	✗	✗	✗	✗	✗	–/–	–

Note: ✓– algorithm is fully supported, ✓!– algorithm is supported, but there is not direct access, ✗– algorithm is not supported, $\mathbb{F}_p$ – prime finite field, $\mathbb{F}_{2^m}$ – binary finite field, $\mathcal{A}$ – affine space, $\mathcal{P}$ – projective space, $\mathcal{T}$ – twisted curve.

3.3.2 Performance Results

This section details the experimental results performance assessment of basic cryptographic functions on main smart card platforms, namely, Java Card, MultOS, Basic Card and .NET Card. An emphasis will be placed on elliptic curve cryptography benchmarks carried out on the different types of smart cards, since our privacy-enhancing schemes, proposed in Sections 4–7 are based on it. The technical specification of tested smart cards is shown in Table 3.4. Selected smart cards represent currently most used cards in practise. Unfortunately, .NET Cards have no support of elliptic curve cryptography, therefore, we have omitted them from our elliptic curve benchmark tests.

Tab. 3.4: Technical specification of tested smart cards.

	J3A081	J3D081	Sm@rtCafe6	Sm@rtCafe5	ZC7.6	ML4	ML3	Gemalto
MCU	P5CD 081	P5CD 081	P5CD 081	P5CDs 080	–	SC23 Z018	SLE78 CLXPM	–
OS	JavaCard 2.2.2	JavaCard 3.0.1	JavaCard 3.0.1	JavaCard 2.2.2	Basic ZC7	MultOS 4.3.1	MultOS 4.3.1	.NET 2.2
ROM	264KB	264KB	264KB	200KB	–	252KB	280KB	80KB
EEPROM	80KB	80KB	80KB	80KB	72KB	18KB	96KB	400KB
RAM	6KB	6KB	6KB	6KB	4.3KB	1.75KB	2KB	16KB

For testing purposes, we took the elliptic curves defined by FIPS [22], SECG [35], the American National Standards Institute (ANSI) [99], NUMS [37], Brainpool [36], the Wireless Transport Layer Security (WTLS) [100] standards. We also considered Barreto-Naehrig pairing friendly curves [42]. All the considered elliptic curves are tested over prime field $\mathbb{F}_p$, since there is only few smart card implementations that support ECs over binary field $\mathbb{F}_{2^m}$. These cards allow small curve sizes (up to 193 bits) for JCs and particular EC sizes for Basic cards. Moreover, there is not support of basic operations over binary field for the smart card platforms. Unfortunately, some smart card implementations do not provide the ECC support as described in their framework specification. The real support depends on the manufacturer itself. The real ECC support is shown in Table 3.5.

Each operation was averaged over 100 executions on all the aforementioned smart cards. Then, the result was sent to the PC for evaluation. Using this methodology, we minimize the impact of communication overhead between PC and a smart card. The overhead depends on the data length, communication interface and other parameters (e.g. conductance or radio frequency field strength, modulation, signal gain, threshold level). If the operation is quite fast, then the delay is more significant in total time [101]. The smart card does not allow us to make the measurements in the Central Processing Unit (CPU) cycles, hence, the results are in ms.

Benchmarks of Elliptic Curve Operations

The basic arithmetic operations on elliptic curves are point addition (`ecAdd`), scalar multiplication (`ecMul`) and point inverse (`ecInv`). We provide the speed of all these operations for Java Card, MultOS and Basic Card.

Tab. 3.5: Elliptic curve support on tested smart cards.

	J3A081	J3D081	Sm@rtCafe6	Sm@rtCafe5	ZC7.6	ML4	ML3
ECC $\mathbb{F}_p$ [bit]	320	320	256	256	544	512	512
ECC $\mathbb{F}_{2^m}$ [bit]	–	–	–	–	211	–	–
ECDSA [bit]	320	320	256	256	544	✗	512
ECDH [bit]	320	320	256	256	544	✗	512
ECIES [bit]	✗	✗	✗	✗	✗	✗	✗
`ecKpGen` [bit]	192	✗	✗	✗	544	✗	512
`ecAdd`	✓!	✓!	✗	✗	✓	✓	✗
`ecMul`	✓!	✓!	✗	✗	✓	✓	✗
`ecInv`	✗	✗	✗	✗	✗	✓	✗

Note: ✓– algorithm is fully supported, ✓!– algorithm is supported only through NXP JCOP API, ✗– algorithm is not supported.

Since Java Card does not support basic EC operations, such as `ecAdd`, `ecMul` or `ecInv`, and `ALG_EC_PACE_GM` and `ALG_EC_SVDP_DH(C)_PLAIN_XY` algorithms are not available in any card present on the market, we had to get in a compromise and use some workaround to perform `ecMul`. The key agreement protocol `ALG_EC_SVDP_DH` up to JC2.2.2 can be used to compute the hash function of an `ecMul` (note that it is not possible to compute `ecMul` in plain). Thus, we had to subtract the time of SHA-1 from our computation. Another possibility is to use `ALG_EC_SVDP_DH_PLAIN` algorithm for JC3.0.1, which, given $P \in E(\mathbb{F}_q)$ and $b \in \mathbb{F}_q$, $Q = (x_q, y_q) = bP$, returns x_q (note that also in this case it is not possible to receive Q). However, currently it is not possible to perform `ecAdd` and `ecInv` on Java Card and receive actual results in plaintext.

In addition to the short Weierstrass form (Equation 2.7), Basic Card allows to use twisted curves, which are defined on projective coordinates (x, y, z). In particular, given a generic curve $y^2 = x^3 + ax + b \bmod p$, its twisted curve is given by $y^2 = x^3 + z^4ax + z^6b \bmod p$, where $F(x, y) = (xZ^2, yZ^3)$ is the isomorphism between them (see [36] more details). MultOS cards allow to use projective coordinates (x, y, z) instead of affine coordinates (x, y). Figure 3.13 depicts the `ecMul` cost for Brainpool curves on Java Card, MultOS and Basic Card. MultOS card are 75% faster than Basic cards (ZC7.6) and 35% faster than the fastest Java cards (J3A081). JC Sm@rtCafe implementations show worse results than JCOP implementations.

A comparison of different ECs is depicted on Figure 3.14. Regarding MultOS implementation, the `ecMul` computation is around 25% faster in the affine space compared with the projective space. There is no significant difference between Random curves and Twisted curves defined by Brainpool standard on ZC7.6 Basic cards.

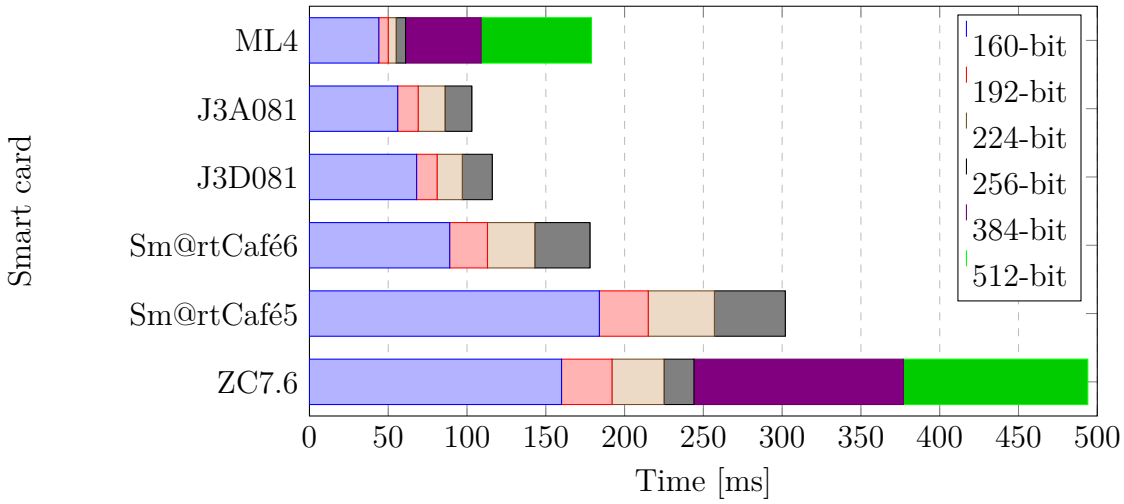

Fig. 3.13: Efficiency of `ecMul` operation on different smart card platforms.

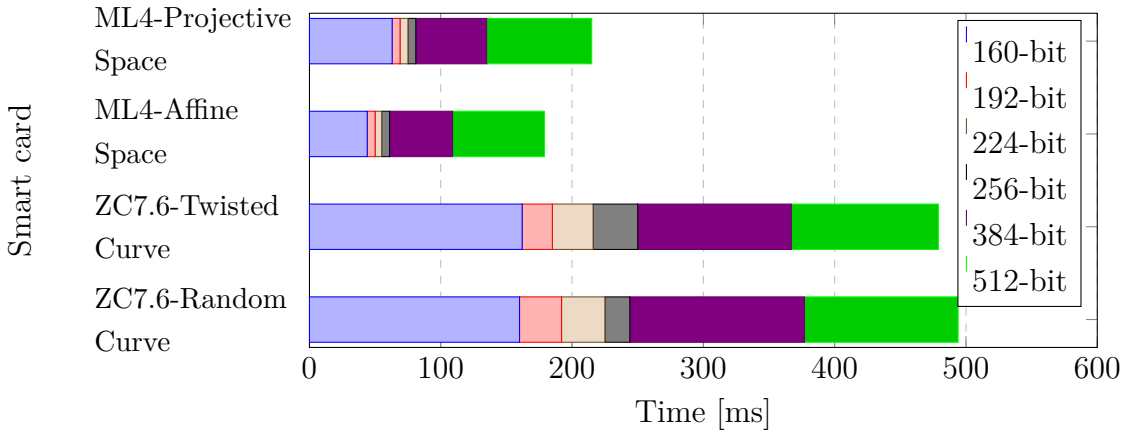

Fig. 3.14: Efficiency of `ecMul` operation based on EC form.

Figure 3.15 shows `ecAdd` and `ecInv` costs on MultOS and Basic Card. For MultOS, `ecAdd` and point doubling require the same time. The `ecAdd` operation is 20% faster on MultOS cards than on Basic cards.

Table 3.6 shows the speed of `ecMul` on Java Card, MultOS and Basic Card. The greater the EC bitlength is, the more time is needed for the computation, as expected. On the contrary, different EC forms with equal EC bitlength present same speed. Only a small difference is visible for JCOP cards, where FIPS and SECG curves show better results.

Benchmarks of Elliptic Curve Protocols

Only some of the ECC protocols are supported on smart cards, namely EC key pair generation (`ecKpGen`), ECDSA signature scheme and ECDH key agreement protocol.

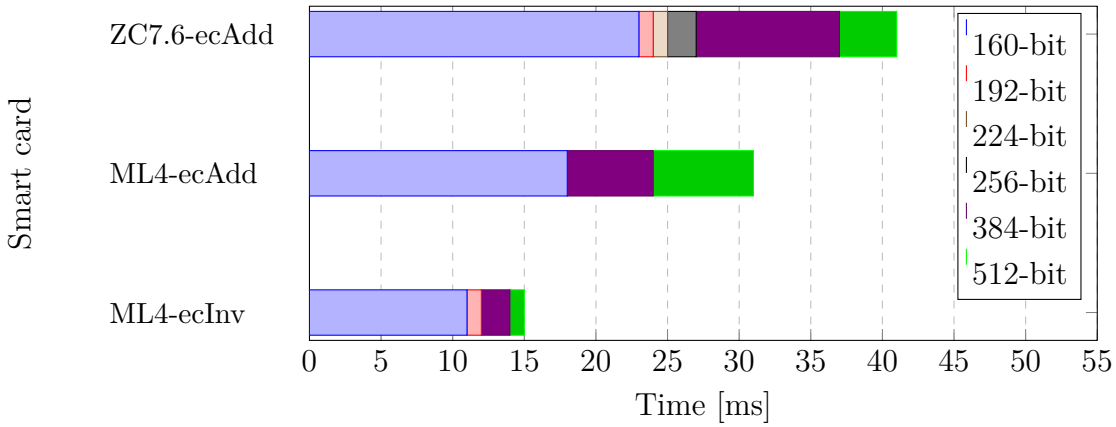

Fig. 3.15: Efficiency of `ecAdd` and `ecInv` operations on different smart card platforms.

The ECC encryption algorithms are not provided by any smart card platform. In this section, we provide a comparison of ECC schemes implementation using FIPS [22] elliptic curves.

The key pair generation function `ecKpGen` is important especially for signature and encryption schemes, where public and private keys need to be generated on the card. This function is supported only for some EC sizes, as it is shown in Figure 3.16. ML3-80K-R1 MultOS smart card supports the `ecKpGen` function but lacks basic arithmetic. On the contrary, ML4 smart card supports basic arithmetic but has no support of ECC protocols. `ecKpGen` works only with 192 bits on Java Card and is supported for different sizes on MultOS and Basics Card. `ecKpGen` is significantly faster on Basics Card compared to MultOS.

ECDH protocol for key agreement purposes is implemented on all the platforms: Java Card, MultOS and Basic Card. Java Card platform achieves the best speed, particularly on the J3A081 smart card, as depicted on Figure 3.16. For large EC-bit length, the results of MultOS ML3 card are rather slow.

Finally, we show the performance of ECDSA on Java Card, MultOS and Basic Card platforms. The speed of the signing and verifying algorithms are depicted on Figure 3.17. As in the ECDH case, the best implementation of ECDSA is provided by Java Card platform and the worst results are provided by the MultOS ML3 card. With J3D081 card, we are able to generate 256-bit signatures in 107 ms and verify them in 113 ms. That is 40% faster for signature generation and 50% faster for signature verification than the MultOS. Moreover, Figure 3.17 shows that in signing ZC.7.6 cards are significantly faster comparing with other platforms, as for computing `ecKpGen`. It is due to the pre-computation of the EC points in EEPROM, which allows to speed up EC operations over the pre-defined curves (FIPS [22] and Brainpool [36]). Others curves are not supported.

Tab. 3.6: Time complexity of operation `ecMul` in *ms* based on different standard, security level and smart card platform.

	Java Card				Basic Card	MultOS
Elliptic Curve	**Sm@rtCafe6**	**Sm@rtCafe5**	**J3D081**	**J3A081**	**ZC7.6**	**ML4**
FIPS, **SECG**, **ANSI**, **WTLS** (Random curves)						
P-192	114	256	74	61	193	50/69*
P-224	143	264	87	74	218	55/75*
P-256	178	310	120	109	256	61/81*
P-384	–	–	–	–	367	109/135*
P-521	–	–	–	–	510	–
SECG (Koblitz curves)						
secp192k1	114	225	73	61	–	50/69*
secp224k1	144	265	87	75	–	55/75*
secp256k1	179	310	101	90	–	61/81*
Brainpool (Random curves)						
bpP160r1	89	192	72	60	160	44/63*
bpP192r1	113	223	85	73	192	50/69*
bpP224r1	143	265	101	90	225	55/75*
bpP256r1	178	307	120	109	244	61/81*
bpP384r1	–	–	–	–	377	109/135*
bpP512r1	–	–	–	–	494	179/215*
Brainpool (Twisted curves)						
bpP160t1	–	–	–	–	162	–
bpP192t1	–	–	–	–	185	–
bpP224t1	–	–	–	–	216	–
bpP256t1	–	–	–	–	250	–
bpP384t1	–	–	–	–	367	–
bpP512t1	–	–	–	–	479	–
NUMS (Random curves)						
numsp256d1	184	309	109	99	–	61/81*
numsp384d1	–	–	–	–	–	108/135*
numsp512d1	–	–	–	–	–	180/215*
BN (Barreto–Naehrig) Pairing-friendly curves						
BN-160	89	192	72	59	–	44/63*
BN-192	113	226	85	73	–	50/69*
BN-224	143	264	101	90	–	55/75*
BN-256	178	307	120	109	–	61/81*
BN-384	–	–	–	–	–	109/135*
BN-512	–	–	–	–	–	180/215*

Note: * – Time for computing operation with EC point defined with *projective coordinates* $\mathcal{P}$ (x, y, z).

Benchmarks of Modular Arithmetic and Hash Functions

In order to have a complex overview of the complexity of the protocol that runs on a real smart card, it is necessary to consider also other relevant operations that

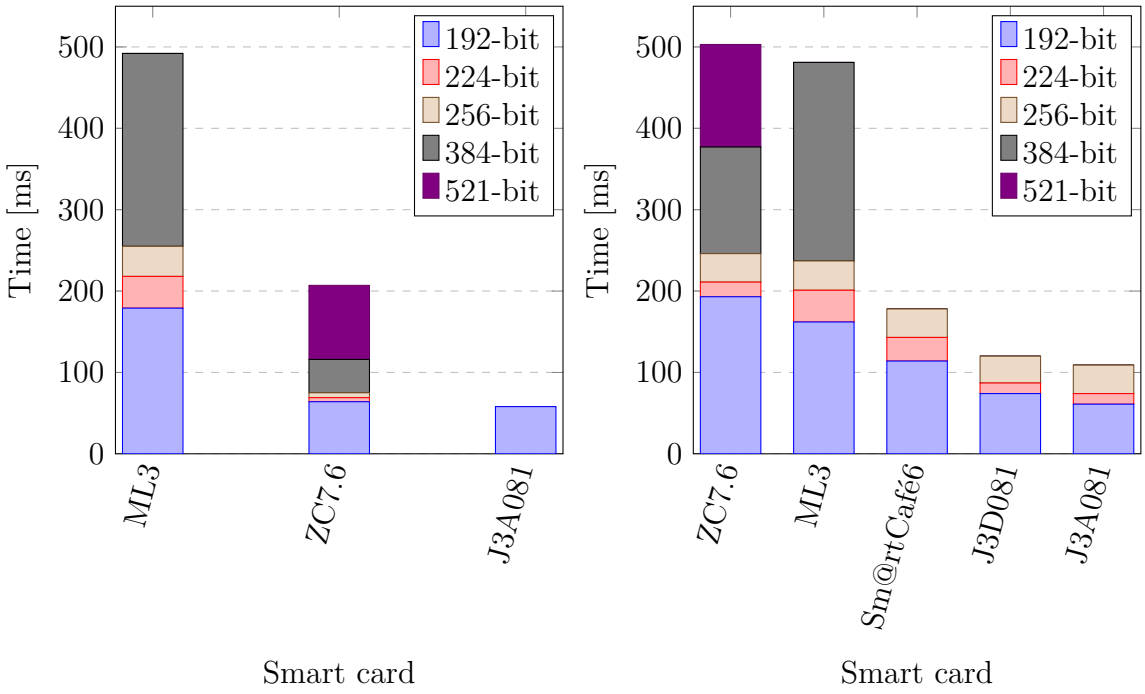

Fig. 3.16: Efficiency of `ecKpGen` algorithm (on the left) and `ECDH` protocol (on the right) on different smart card platforms, NIST-P.

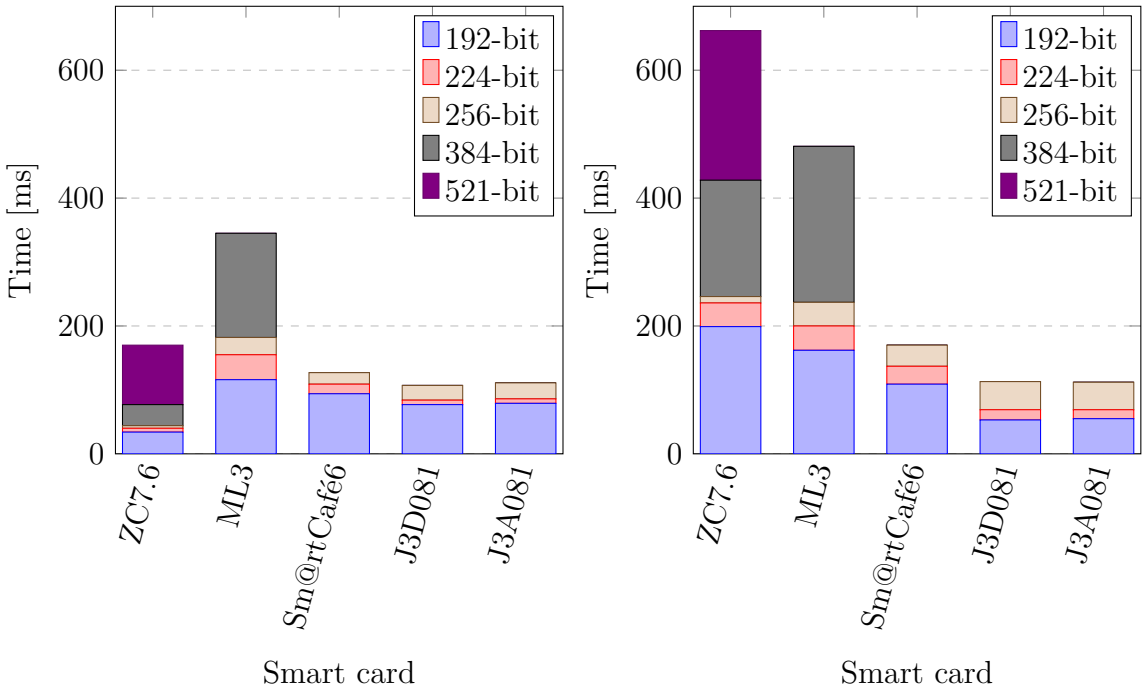

Fig. 3.17: Efficiency of `ECDSA Signing` (on the left) and `Verifying` (on the right) algorithms on different smart card platforms.

are frequently used. For example, most of the privacy-preserving protocols are based on zero-knowledge proofs, where additional operations such as random number generation and multiplication and addition of large numbers are used. Moreover, non-iterative protocol versions are based on Fiat-Shamir heuristic [64], which uses hash functions.

Basic arithmetic operations are directly supported only by Basic Cards and MultOS Cards and take only few tens of ms, see Figure 3.18. These operations on Java Card and .NET Card are not directly supported, therefore a software implementation must be used. We provide results of our multiplication operation implementation using the RSA tunnel method [102].

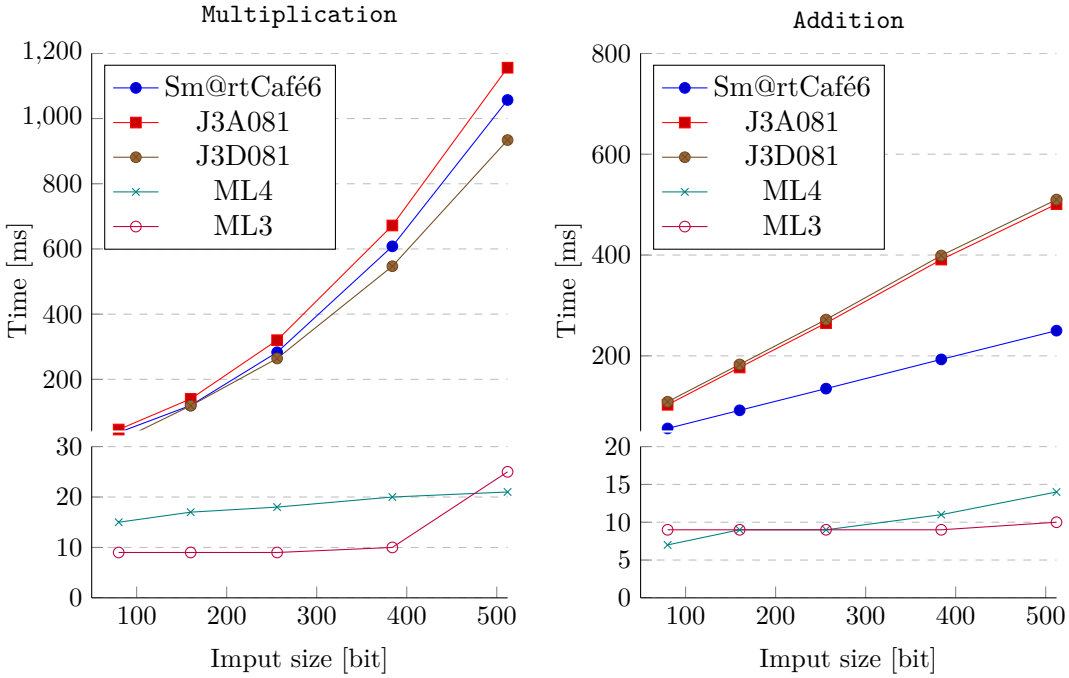

Fig. 3.18: Modular multiplication and addition cost on smart card.

Figure 3.19 shows efficiency of random number generation for different sizes of numbers. The time consuming takes only few tens of ms (for small bitlengths up to 512-bit). The best results are achieved with Sm@rtCafé6 smart card. In order to see the advantages of elliptic curves cryptography, we provide time consuming of modular exponentiation for $n = 2048$ bits, which is comparable to security strength of 224-bit elliptic curve, see Figure 3.19. The operation `ecMul` is pretty faster than `Modular Exp.`, however, the value of the difference depends on the smart card implementation. Whereas in case of Sma@rtCafé6 there is a negligible difference, in case of J3A081 `ecMul` is ca. 6x faster than `Modular Exp`. The difference became bigger with the growth of the security strength.

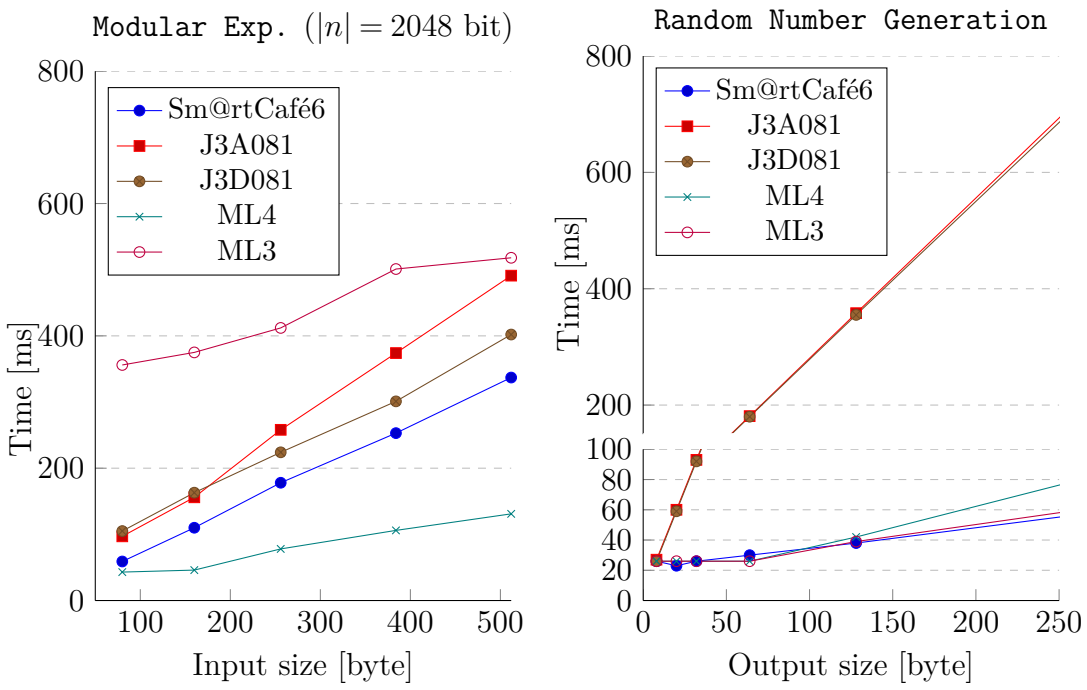

Fig. 3.19: Modular exponentiation ($|n| = 2048$ bit) and random number generation.

The choice of the hash function has significant impact on a protocol speed as well. The algorithm SHA-1 shows ca. 2x better results than SHA-256, see Figure 3.20 for more details.

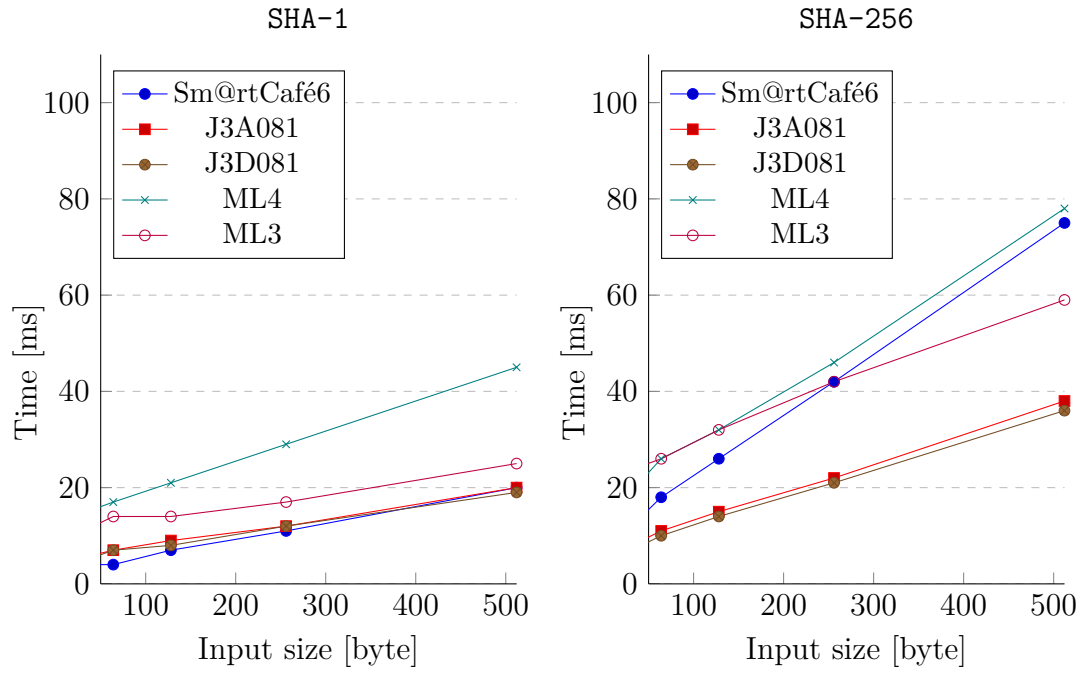

Fig. 3.20: Message digest based on hash function SHA-1 and SHA-256.

4 Multi-Device Authentication with Strong Privacy Protection

The content of this chapter have been published in impact factor journal paper [13].

4.1 Introduction

Privacy-enhancing technologies constitute a significant part of contemporary cryptography. Modern cryptographic protocols allow privacy-enhanced storing of sensitive data and its processing by cloud services, private information retrieval, or, for example, authentication based on personal attributes, instead of user identifiers. The increasing intensity of research into privacy is supported by national programs and strategies, in particular in US [7] and EU [10]. While most of the novel schemes are aimed at electronic services, the domain of physical access control is rather neglected. We still use traditional locks, tourniquets and classical card-based access control mechanisms to manage physical access to our premises. But with the increasing computational power of the programmable smart cards, massive expansion of various personal electronic devices and the capabilities in RFID communication of our smart phones, we can expect penetration of privacy-enhancing technologies also to the area of physical access control. In particular, in mass applications like public transportation, e-ticketing, e-passports and eIDs, the benefits of controlling physical access using electronic devices with advanced cryptographic protocols are very appealing.

In this chapter, we propose and experimentally evaluate a novel cryptographic scheme that particularly addresses two phenomenons of contemporary cyberspace: lack of user privacy and ubiquitous presence of many personal devices (phones, smart cards, RFID tags, bluetooth dongles, smart watch, etc.) that can be leveraged for stronger authentication and more reliable access control.

In particular, we focus on safety applications in which the users wear multiple safety equipment, such as helmets, harnesses, boots, protective suits, etc., each with attached programmable RFID tag capable of wireless communication. A user is granted access to (potentially dangerous) premises only if all his equipment is present. In existing systems, the presence of the protective equipment is checked simply by scanning the identifiers using RFID readers. Such an approach is neither secure (identifiers can be counterfeited), nor privacy friendly (identifiers can be traced, behavioral profiles can be created, etc.).

We propose a novel cryptographic scheme for multi-device authentication that is tailored to physical access control systems where the user must prove not only his

own identifier, but also *many other auxiliary identifiers* stored on separate devices. In addition, the authentication sessions must support all the key privacy-enhancing features, i.e., the access control process must be *anonymous* (i.e., a user must prove that he belongs to a group of authorized users, but without releasing his concrete identity), *unlinkable* (all the sessions of a single user cannot be linkable to a profile) and *untraceable* (system administrators must be unable to trace honest users in the system). On the other side, the scheme must provide efficient means for revocation and identification of malicious users. In our cryptographic system, we provide all the required features that are often contradictory and completely unavailable in existing schemes (in particular, the presence of many identifiers vs. anonymity; the untraceability and strong cryptographic security vs. efficiency on RFID tags and stickers).

In our scheme, users can be granted an access to premises upon proving the presence of particular devices in their proximity (e.g., the safety equipment) or personal attributes (age, membership, citizenship, etc.). The access control process may[1] proceed in a fully private manner, without disclosing user identity or being traceable in the system.

4.2 Related Work

Most of the existing practical physical access control systems are based on the following technologies [103]: NXP's Mifare and DESfire; HID's Prox and iClass; and Legic Prime and Advant. NXP's Mifare Classic, introduced in 1994, is a very popular technology used in physical access control systems. Although very old and insecure, the technology is still used in many applications, even those security sensitive. The authentication protocol is based on a unique 4B card identifier User IDentifier (UID). In some implementations, the card just reveals UID to the terminal without any authentication protocol. In that case, UID can be easily eavesdropped and used by an attacker for impersonation. In other implementations, a simple authentication protocol is used but is considered insecure due to many existing practical attacks [104, 105, 106] on the encryption algorithm CRYPTO1. The insufficient security of the CRYPTO1 algorithm used in the Mifare Classic made NXP improve the cryptographic protection and release Mifare DESFire. The old encryption algorithm was replaced by Data Encryption Standard (DES) and 3DES algorithm. The authentication protocol was further improved in Mifare DESFire EV1 which supports the AES encryption algorithm [107]. The protocol itself remained without any major

[1] The extent of privacy-enhancing features can be initially set by the administrator. If required, identification or user tracing may be enforced by the access control system.

changes. However, even Mifare DESFire was successfully attacked, although the attacks [108, 109] were aimed on the implementation, not cryptographic weaknesses. The HID Prox technology contains no cryptographic protection. HID iClass employs an authentication protocol based on the 3DES algorithm, but attacks on this protocol are available [110]. Legic Prime has weak proprietary cryptographic protection [111]. Legic Advant is protected by symmetric block algorithms (DES [112], 3DES, AES). None of the major commercial technologies provide any protection of privacy.

With the introduction of the first attribute-based credential schemes, such as the Idemix [113], U-Prove [87] and HM12 [88], the variants for physical access control systems also started to appear. The U-Prove scheme was implemented on MultOS smart cards [91]. The user is able to prove his attribute in less than 1 s using this implementation. However, the unlinkability property cannot be provided by the cryptographic design of the protocol. The Idemix was also implemented on the MultOS smart card platform [1], with ca. 1 s needed to generate the attribute proof. The pilot implementation of the HM12 scheme using MultOS ML3 smart cards [96] required around 2.4 s in total to generate and verify the proof, including the communication overhead. No testing was done on multiple devices because the distributed proof is not supported by these schemes.

Many types of personal and wearable devices forming the so-called Internet of Things have appeared recently. Authentication issues have been solved by different techniques on these devices. Xu and Weitao [114] propose biometric authentication using wearables with face recognition using smart-glass and gait recognition using smart-watch.

Riva et al [115] combine multiple sources of authentication data, which is close to our approach. However, all these schemes are using mainly biometric authentication factors. Cha et al [116] present a simple model for two device authentication for micro-payment systems using mobile and wearable devices. Nevertheless, their proposal lacks details and concrete cryptographic functions. Butun et al [117] address multilevel authentication issue in cloud computing. Gonzalez-Manzano et al [118] present an access control mechanism for cloud-based storage service access by using a set of devices. However, their scheme is based on symmetric cryptography, thus does not provide non-repudiation. Hajny et al [119] use many wearable and IoT devices to do the authentication process. However, the scheme misses privacy-enhancing properties, because each user is uniquely represented by his/her public key.

In summary, there are several authentication solutions that involve IoT devices. However, there are only very few papers focusing on multi-device authentication. Currently, none of the proposals is provably secure and supports the

privacy-enhancing features. Furthermore, most of the schemes remain only theoretic.

Our Contribution

The cryptographic scheme presented in this chapter takes a novel approach for the access control based on rather the presence of multiple devices in user's proximity than the direct verification of user identifiers. The novel approach has two key benefits: it significantly improves the privacy protection of users and allows the authentication based on the presence of many low-performance devices. Our scheme is the first practical proposal with implementation results that combines strong security, all standard privacy-enhancing features and efficiency:

- **Provable security**: all algorithms are provably secure, based on primitives with rigorous formal proofs.
- **Multi-device authentication**: the scheme allows user authentication based on the presence of many personal devices.
- **Anonymity**: the scheme allows authentication based on anonymous proofs of knowledge of private user and/or device identifiers.
- **Unlinkability**: the scheme prevents creating user behavior profiles based on the authentication sessions linking.
- **Untraceability**: the scheme prevents any entity from tracing users (or their devices).
- **Efficiency**: the authentication protocol is fast on constrained user devices (i.e., smart cards) and embedded verification terminals.
- **Revocation and identification**: the proposed scheme is compatible with major revocation and identification schemes [120, 121, 66] for attribute-based credentials.

We provide not only the cryptographic description and security proofs of our scheme, but also provide practical implementation results based on benchmarks on RFID devices and an embedded hardware terminal. These results prove that the scheme can be practically implemented on existing off-the-shelf devices.

4.3 Cryptographic Design

At first, we define the formal requirements on the authentication scheme. Then, we define the algorithms and entities in the scheme. At last, we present the concrete instantiation of the privacy-enhanced multi-device authentication scheme based on the wBB signatures described in the Chapter 2.

4.3.1 Requirements

We require the scheme be secure, private and efficient. Below, the requirements are described in details.

Security Requirements

- *Completeness*: registered users must be accepted by the `Authenticate` protocol.
- *Soundness*: unregistered users must be rejected by the `Authenticate` protocol.
- *Zero-Knowledge*: the `Authenticate` protocol transcript must be simulatable without the knowledge of identifiers, so that provably releases no sensitive information.

Privacy Requirements

- *Anonymity*: users must be able to prove the knowledge of their identifiers anonymously, i.e. without disclosing them.
- *Untraceability*: user authentication sessions must be untraceable by all system entities, including registrars.
- *Unlinkability*: all single user's authentication sessions must be mutually unlinkable.

Efficiency Requirements

- *Readiness for RFID devices*: the scheme must be fast on constrained devices, such that smart cards. No operations, that are unavailable on RFID devices (such as bilinear pairings), can be performed by user's algorithms.

4.3.2 General Architecture

In this section, we define the algorithms of our scheme. The communication pattern is depicted in Figure 4.1 and employs the registrar (i.e., a central server that manages users and their equipment), users (i.e., user devices such as smart cards or smart phones), terminals (i.e., embedded devices with RFID readers typically attached next to doors) and tags (i.e., devices that need to be present during authentication and access control, typically safety equipment with programmable RFID sticks, such as the helmet, respirator or harness).

$(par) \leftarrow$`Setup`$(1^\kappa, n)$: the algorithm is run by the registrar. It inputs the security parameter κ and the maximum number of tag classes n (i.e., helmets, harnesses, boots, etc.). The algorithm outputs the public system parameters par.

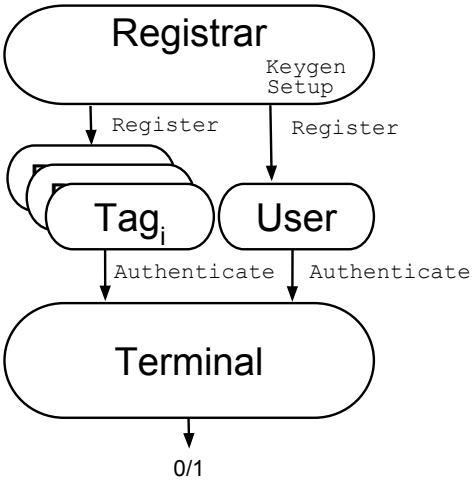

Fig. 4.1: Architecture of multi-device authentication with privacy protection.

$(sk_r, pk_r) \leftarrow$`Keygen`(par): the algorithm is run by the registrar. On the input of public system parameters par, it generates its private key sk_r and public key pk_r. The registrar distributes the public key to all other entities.

$(\langle ID_i, \sigma_i \rangle_{i=1}^{n}, ID_u, \sigma_u) \leftarrow$`Register`$(par, sk_r, pk_r)$: the algorithm is run by the registrar. On the input of system parameters and its keypair, the registrar generates the tags' identifiers ID_i with corresponding signatures σ_i and user's identifier ID_u with a corresponding signature σ_u. The tag identifiers and signatures are securely delivered to tags and the user identifier and signature are delivered securely to the user device.

$(0/1) \leftarrow$`Authenticate`$(par, \langle ID_i, \sigma_i \rangle_{i=1}^{n}, ID_u, \sigma_u, pk_r)$: the cryptographic protocol is run jointly by the user device, tags and the terminal. It inputs system parameters, registrar's public key, private identifiers and corresponding signatures, and returns 1 iff signatures and IDs are valid, or 0 otherwise.

4.3.3 Cryptography Specification

In this section, we present the concrete instantiations of cryptographic algorithms defined in Section 4.3.2. We use the wBB signature scheme to certify the identifiers of tags and users in the `Register` algorithm and interactive proofs of knowledge to prove the knowledge of respective signatures and identifiers in the `Authenticate` protocol. We use the Camenisch-Stadler notation [17] to describe the proof of knowledge protocols.

`Setup`

$(par) \leftarrow$`Setup`$(1^\kappa, n)$: the algorithm inputs the security parameter κ and the maximum number of tag classes n. It generates the bilinear group with parameters $par = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, \ldots, g_n, g_u \in \mathbb{G}_1, g_{\mathbf{2}} \in \mathbb{G}_2)$ satisfying $|q| = \kappa$.

`Keygen`

$(sk_r, pk_r) \leftarrow$`Keygen`(par): the algorithm inputs the public parameters par, selects random registrar's private keys $sk_r = (sk_0, sk_1, \ldots, sk_n, sk_u) \xleftarrow{\$} \mathbb{Z}_q^*$ and computes the public keys $pk_r = (pk_0 \leftarrow g_{\mathbf{2}}^{sk_0}, pk_1 \leftarrow g_{\mathbf{2}}^{sk_1}, \ldots, pk_n \leftarrow g_{\mathbf{2}}^{sk_n}, pk_u \leftarrow g_{\mathbf{2}}^{sk_u})$. It outputs the private keys as registrar's private output and the public key as the public output.

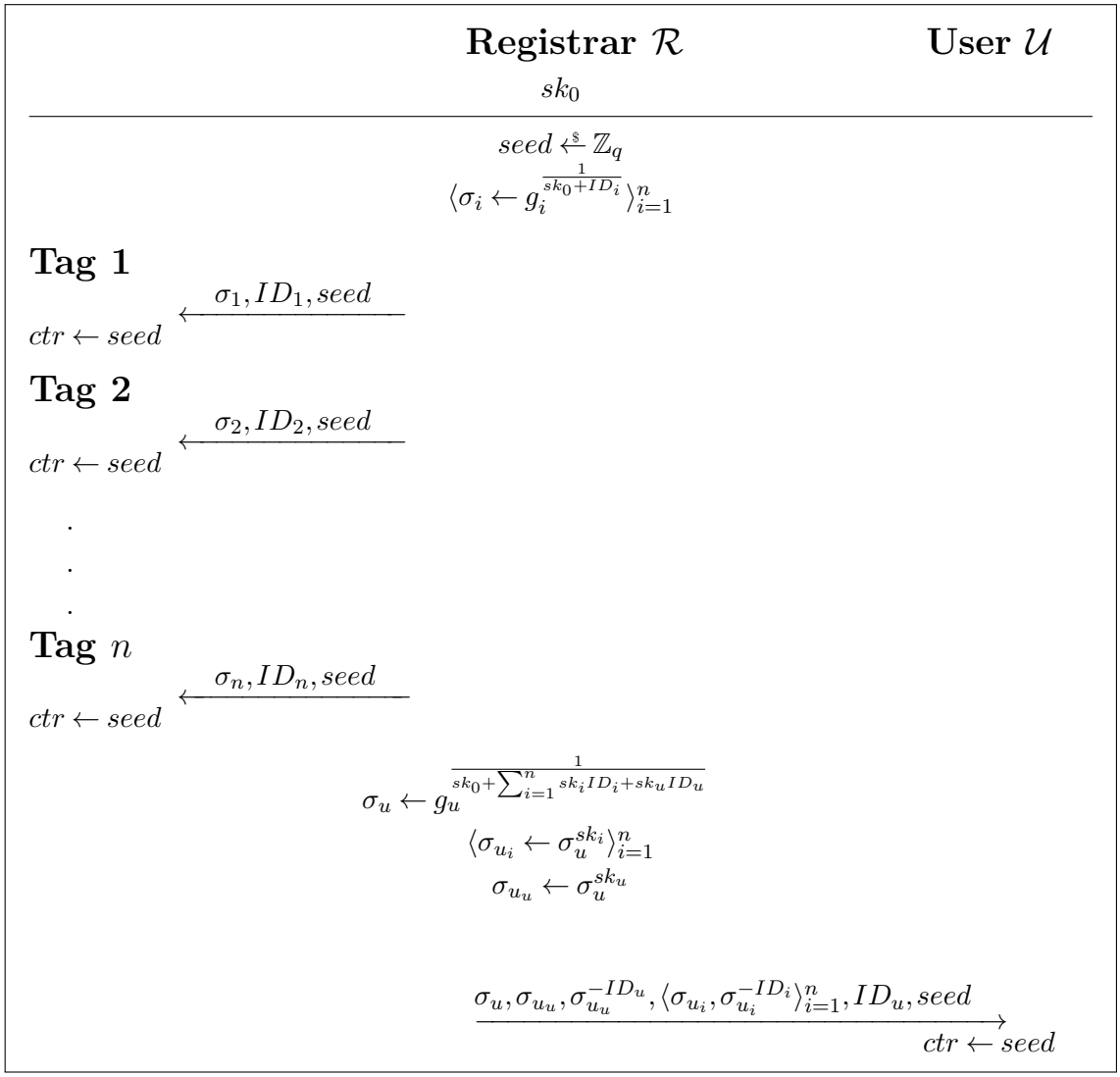

Fig. 4.2: `Register` protocol.

`Register`

$(\langle ID_i, \sigma_i \rangle_{i=1}^{n}, ID_u, \sigma_u) \leftarrow$`Register`$(par, sk_r, pk_r)$: the algorithm inputs the registrar's keys and public parameters, randomly selects tag and user identifiers (ID_1, ..., ID_n, ID_u) $\xleftarrow{\$} \mathbb{Z}_q$ and computes the wBB signatures $(\sigma_1, \ldots, \sigma_n)$ on tag identifiers $(ID_1, \ldots, ID_n)$ and the aggregated user signature σ_u and auxiliary values $\langle \sigma_{u_i}, \sigma_{u_i}^{-ID_i} \rangle_{i=1}^{n}$, $\sigma_{u_u}, \sigma_{u_u}^{-ID_u}$ that allow the construction of efficient proofs of knowledge in the `Authenticate` protocol. The algorithm outputs the tag identifiers and corresponding signatures as a private output to tags. The user identifier, the aggregated signature and auxiliary values are outputted to the user as a private output. Both tags and the user receive the initial *seed* required for the synchronization of the zero-knowledge proofs as a private input. The algorithm is depicted in Figure 4.2.

`Authenticate`

$(0/1) \leftarrow$`Authenticate`$(par, \langle ID_i, \sigma_i \rangle_{i=1}^{n}, ID_u, \sigma_u, pk_r)$: the algorithm is distributed among the user, terminal and tags that inputs the identifiers and respective signatures and outputs 1 iff 1) all signatures are valid and created by the registrar, and 2) all identifiers of the user are present and signed. Otherwise it outputs 0. The protocol is a distributed proof of knowledge of wBB signatures where the tags prove that they know their identifiers and corresponding signatures (without actually revealing them) and, at the same time, the user proves that he has an aggregated signature on all his tag identifiers, plus his own identifier. As the user does not know the tag identifiers, all tags must be present and participate on the proof construction. As a result, the user is able to anonymously, untraceably and unlinkably prove his valid registration by the registrar and the presence of all his tags, i.e., the safety equipment. The protocol is depicted in abstract CS notation in Figure 4.3 below. We also provide the full description in Figure 4.4 in Section 4.5 focused on implementation.

4.4 Security Analysis

In this section, the security of the aforementioned scheme is analysed. Since the registrar issues the wBB signatures to tags and users in the `Register` algorithm. Then, the user and tags prove the knowledge of such signatures to the terminal using the distributed zero-knowledge proofs in the `Authenticate` protocol.

Lemma 1. *The weak Boneh-Boyen signatures are unforgeable against a weak chosen message attack under the q-Static Diffie-Hellman assumption [65].*

The Lemma 1 is proven in [65].

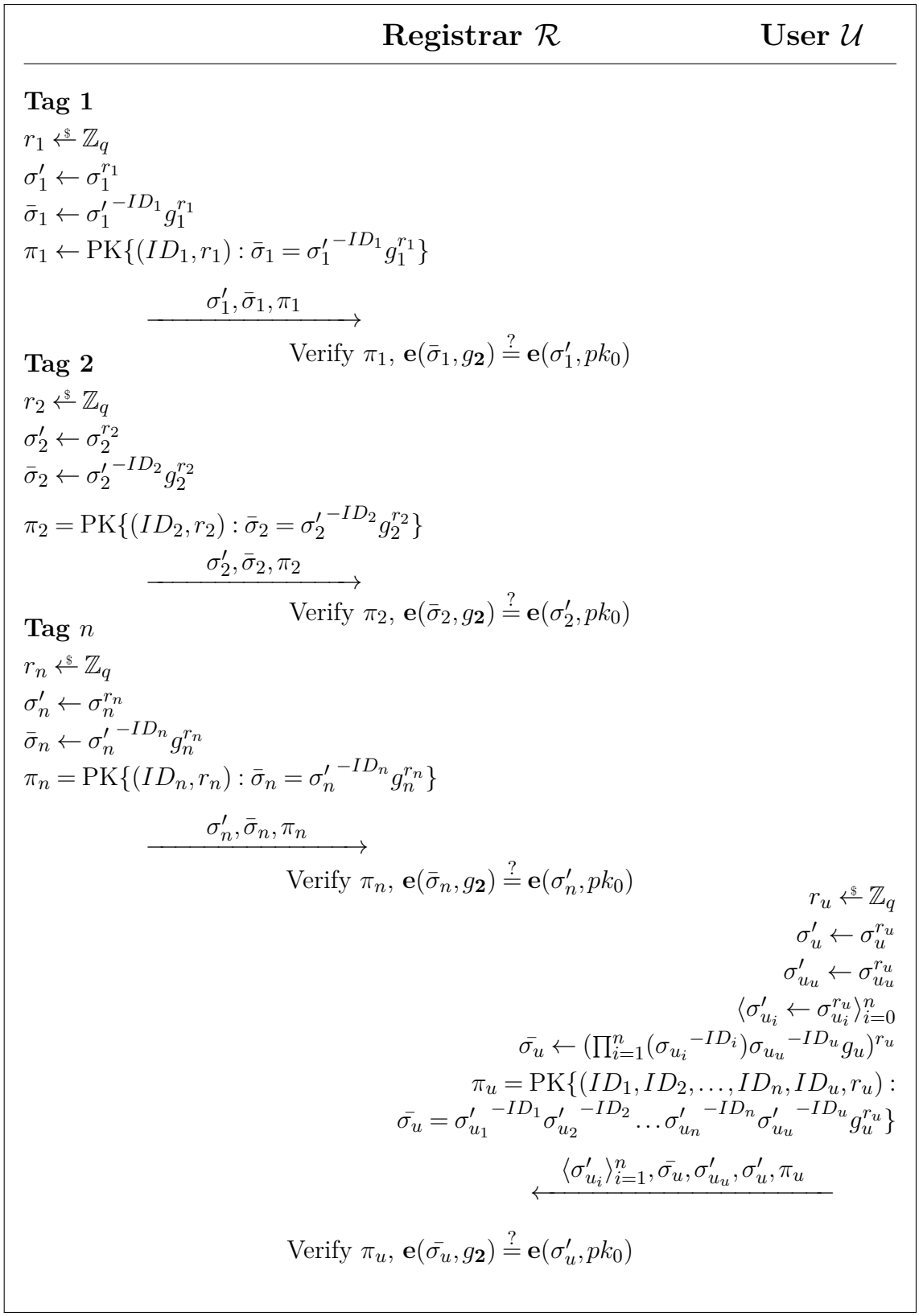

Fig. 4.3: `Authenticate` protocol in CS notation.

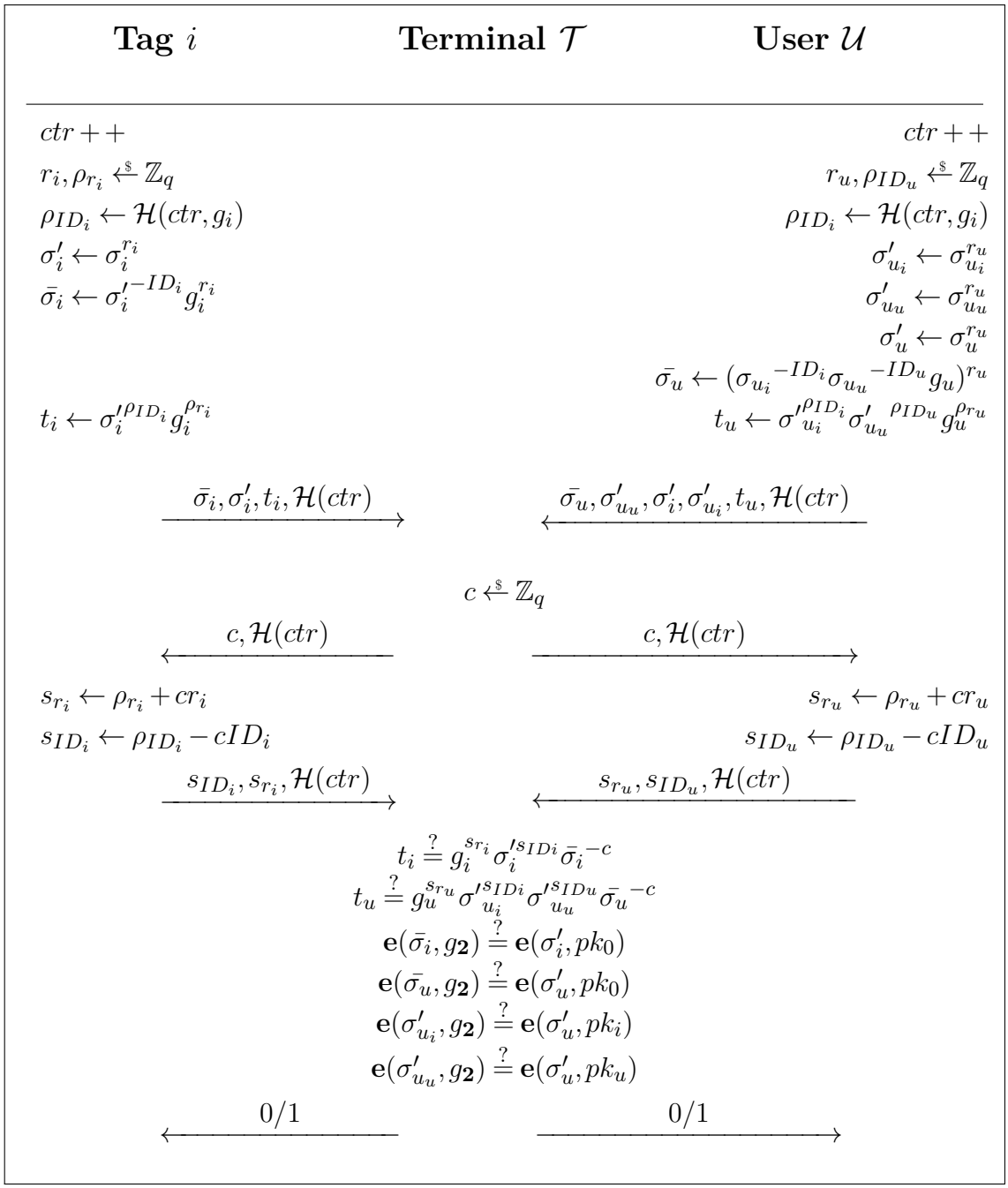

Fig. 4.4: `Authenticate` protocol in full notation for i^{th} tag.

Lemma 2. *The protocol presented in Figure 4.4 is complete, sound and zero-knowledge.*

We construct the proof for a tag i using the standard proving technique for zero-knowledge protocols. For other devices and the user, the proof is constructed analogically.

Proof. Completeness: honest users pass the terminal's check.

$$t_i = g_i^{s_{r_i}} \sigma_i'^{s_{IDi}} \bar{\sigma_i}^{-c} \tag{4.1}$$
$$= g_i^{\rho_{r_i}} g_i^{cr_i} \sigma_i^{r_i s_{ID_i}} \sigma'^{ID_i c}_i g_i^{-cr_i} \tag{4.2}$$
$$= g_i^{\rho_{r_i}} g_i^{cr_i} \sigma_i^{r_i \rho_{ID_i}} \sigma_i^{-r_i c ID_i} \sigma_i^{r_i ID_i c} g_i^{-cr_i} \tag{4.3}$$
$$= g_i^{\rho_{r_i}} \sigma_i'^{\rho_{ID_i}} \tag{4.4}$$

$$\mathbf{e}(\bar{\sigma_i}, g_{\mathbf{2}}) = \mathbf{e}(\sigma_i', pk_0) \tag{4.5}$$
$$\mathbf{e}(\sigma_i^{-ID_i r_i} g_i^{r_i}, g_{\mathbf{2}}) = \mathbf{e}(\sigma_i', g_{\mathbf{2}}^{sk_0}) \tag{4.6}$$
$$\mathbf{e}(g_i^{\frac{-ID_i r_i + r_i(sk_o + ID_i)}{sk_0 + ID_i}}, g_{\mathbf{2}}) = \mathbf{e}(\sigma_i', g_{\mathbf{2}}^{sk_0}) \tag{4.7}$$
$$\mathbf{e}(g_i^{\frac{r_i sk_o}{sk_0 + ID_i}}, g_{\mathbf{2}}) = \mathbf{e}(\sigma_i', g_{\mathbf{2}}^{sk_0}) \tag{4.8}$$
$$\mathbf{e}(\sigma_i'^{sk_0}, g_{\mathbf{2}}) = \mathbf{e}(\sigma_i', g_{\mathbf{2}}^{sk_0}) \tag{4.9}$$

□

Error probability: if implemented correctly, the user will be always accepted.

Proof. Soundness: only registered users pass terminal's check.

Assume a user who is not registered (i.e., does not know the identifier ID_i) and passes the terminal's check for two different challenges c and c' with two different responses s and s':

$$t_i = g_i^{s_{r_i}} \sigma_i'^{s_{IDi}} \bar{\sigma_i}^{-c} \tag{4.10}$$
$$t_i = g_i^{s'_{r_i}} \sigma_i'^{s'_{IDi}} \bar{\sigma_i}^{-c'} \tag{4.11}$$

and we get:

$$\bar{\sigma_i}^{c-c'} = g_i^{s_{r_i} - s'_{r_i}} \sigma_i'^{s_{IDi} - s'_{IDi}} \tag{4.12}$$

and therefore:

$$\bar{\sigma_i} = g_i^{\frac{s_{r_i} - s'_{r_i}}{c-c'}} \sigma_i'^{\frac{s_{IDi} - s'_{IDi}}{c-c'}} \tag{4.13}$$

Thus the user can efficiently compute both the randomizers $r_i = \frac{s_{r_i} - s'_{r_i}}{c - c'}$ and the identifier $ID_i = \frac{s_{IDi} - s'_{IDi}}{c - c'}$ and we reached the contradiction to our original assumption.

□

Error probability: the attacker will pass the verification check if he can predict the challenge c. The probability of soundness error is thus $P = 2^{-|c|} = 2^{-q} = 2^{-224}$, which is negligible. With an expected rate of 100 ms per challenge, the expected time of breach is 4×10^{58} years.

Proof. Zero-Knowledge: the protocol releases no private information, i.e., there exists a zero-knowledge simulator M_V^*. Using the public parameters and the public key $(\bar{g}, \bar{g}^x)$[2], the simulator chooses randomly and uniformly $(s_{r_i}, s_{ID_i}, r, c) \xleftarrow{\$} \mathbb{Z}_q$, computes $\sigma'_i = \bar{g}^r$, $\bar{\sigma}_i = (\bar{g}^x)^r$, $t_i = g_i^{s_{r_i}} \sigma_i'^{s_{IDi}} \bar{\sigma}_i{}^{-c}$ and outputs the proof $\pi = (\bar{\sigma}_i, \sigma'_i, t_i, \mathcal{H}(r), c, (s_{r_i}, s_{ID_i}))$. The simulated transcript is computationally indistinguishable from the real run of the protocol. □

Error probability: the attacker can try to guess the randomizers $r_i, \rho_{r_i}, r_u, \rho_{r_u}$ and break the discrete logarithm assumption. The probability is $P = 2^{-q} = 2^{-224}$ for each device, which is negligible. With an expected rate of 10 ms per computing the guess (the exponentiation), the expected time of breach is 4×10^{57} years.

As a result of the zero-knowledge property and randomization of all signatures, the protocol is also *anonymous*, *untraceable* and *unlinkable*.

4.5 Implementation and Performance Analysis

The `Authenticate` protocol has been implemented as a standard 3-way interactive zero-knowledge proof of knowledge protocol described in Section 2.4. We use a parallel composition with one challenge and one response for all tags of a user to construct an AND proof for both tag and user signatures. The `Authenticate` protocol for i^{th} tag is fully specified in Figure 4.4.

To keep user devices synchronized, we use a counter that is initialized by a seed generated by the registrar. In the beginning of each session, the counter increments. To avoid loosing synchronisation, the hashed counter is broadcasted by the terminal so that the devices can compare it with their actual counter value (and with, e.g., 10 next pre-computed values) and sync in case their counter is behind. The hashed

[2]We follow the proof presented in [66] that allows the simulator to input an auxiliary public key $(\bar{g}, \bar{g}^x) : \bar{g} \xleftarrow{\$} \mathbb{Z}_q$ from the registrar.

counter also serves as the session identifier, thus is present in all three steps of the protocol.

In the first step of the protocol, the tag generates randomizers $r_i, \rho_{r_i}, \rho_{ID_i}$, computes randomized signatures $\sigma_i', \bar{\sigma}_i$ and computes the commitment to randomizers t_i. The randomized signatures, commitment to randomizers and hashed randomizers are sent to the terminal. In the second step, the terminal randomly selects its challenge c and sends it to all tags and devices, together with the obtained hash. In the third step, the tag computes their answers s_{r_i}, s_{ID_i} of the zero-knowledge protocol.

After receiving the answers, the terminal is able to verify that the tag knows a valid signature and a corresponding tag identifier with respect to registrar's public key pk, without actually learning any user- or tag-identifying values. The proof construction for the user is the same with the exception that the answers containing tag IDs are omitted, because the terminal makes use of the values received by the devices. Instead of proving tag IDs, the user proves the knowledge of his own user ID.

4.5.1 Performance Analysis

The scheme was designed to be practical and fast on constrained RFID devices, such as smart cards and programmable RFID tags. Therefore, the bilinear pairings, which are the most computationally complex operations in our algorithm, are only computed in the terminal which normally has more resources than user device. The second most complex operation is the exponentiation (implemented as scalar multiplication of an elliptic-curve point) and it is reduced to a minimum. The user device needs $(5 + 2d)$ exponentiations to construct a "user proof" with d personal tags. Each tag must compute 5 exponentiations to generate a "tag proof". However, our implementation uses only 4 exponentiations, since the value $({\sigma_i}^{-ID_i} g_i)$ is precomputed within a `Register` protocol and is used for the randomized signature $\bar{\sigma_i} = {\sigma_i'}^{-ID_i} g_i^{r_i} = ({\sigma_i}^{-ID_i} g_i)^{r_i}$ construction. The complexity of the other operations (random number generation, addition and multiplication) are only minor, compared to pairings and exponentiations. In order to verify the proof, the terminal must compute $(4 + 4d)$ bilinear pairings and $(4 + 3d)$ exponentiations.

We provide performance measurement of crucial operations on common devices, which are widely used in the access control applications, i.e. a smart card, smart phone, smart watch (as user devices), a custom-built RFID terminal with ARM or Intel CPU and programmable RFID tags (as RFID tags attached to safety equipment). The hardware and software specification of all the devices is presented in Table 4.1.

Tab. 4.1: Specification of tested devices.

	Type	**CPU/MCU**	**OS**	**RAM**
Tag	Smart Card	SC23Z018	MultOS 4.3.1	1.75 KB
User	Smart Card	SC23Z018	MultOS 4.3.1	1.75 KB
User	Phone	Kirin 655	Android 7.0	3 GB
User	Watch	ARM Cortex-A7	Android 7.0	768 MB
Terminal	Pi 3	ARM Cortex-A53	Raspbian 9.3	1 GB
Terminal	PC	Intel i7-7700	Debian 8.6	16 GB

Note: Tag – programmable RFID stick, User – user device, Terminal – terminal, Smart Card – ML4, Phone – HUAWEI P9 Lite 2017, Pi 3 – Raspberry Pi 3 Model B, Watch – HUAWEI Watch 2

The testing scenario is depicted in Figure 4.5. The user needs to hold a wearable device, such as a smart phone (HUAWEI P9 Lite 2017), a smart card (MultOS Card) or smart watch (HUAWEI Watch 2) and some safety equipment, such as helmets, harnesses, boots, protective suits, each of them with a programmable RFID tag attached. The tag is equipped with a programmable chip SC23Z018 with MultOS 4.3.1 operation system. The proofs are collected and verified by a terminal. We use Raspberry Pi 3 to represent the terminal. In another scenario, PC (Intel i7-7700 CPU, 16 GB RAM) acts as a central authentication server representing the case of a centralized access control system. The system uses RFID communication between tags and a terminal, and NFC or Bluetooth Low Energy (BLE) communication between a terminal and a user device.

The performance of critical operations and the estimation of the running time of the `Authenticate` protocol with one RFID tag and one user device are presented in Table 4.2. In addition, we provide measurement of the selected devices where we consider different elliptic curves types, in particular type A and D. Both curves satisfy the NIST key recommendation for 80-bit security strength [2]. The performance is measured in milliseconds[3] and the values are an average of 10 measurements, excluding communication overhead. For the implementation of EC operations, the PBC library [56] was used on the terminal and jPBC [84] library on Android devices. Native assembler code was used to perform operations on the MultOS smart card.

The proposed authentication scheme can by used in many types of access control

[3]The measurement of clock cycles is unavailable on the smart card platform.

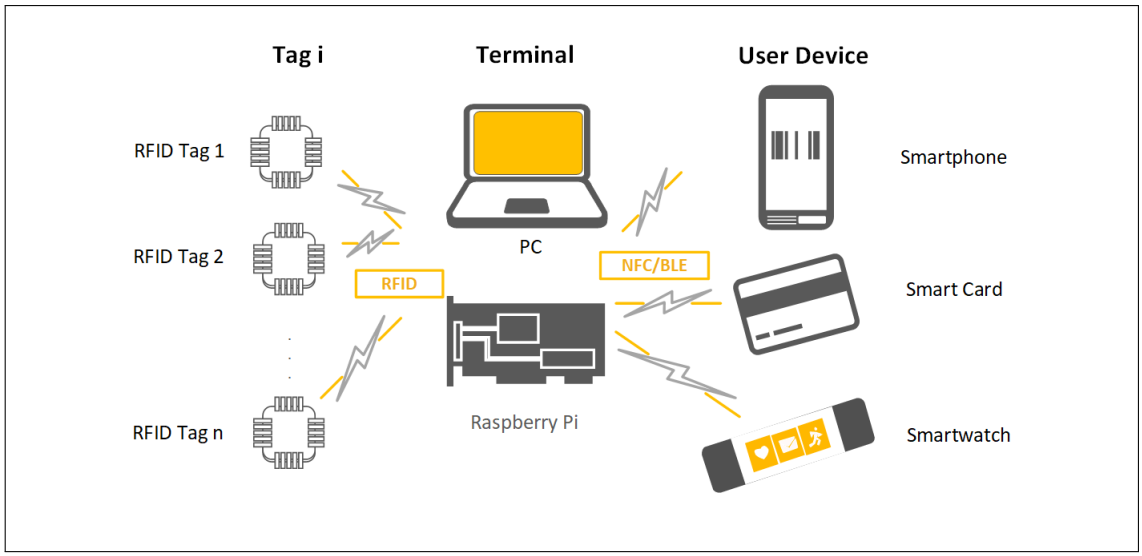

Fig. 4.5: Tested scenario.

Tab. 4.2: Benchmark results based on elliptic curve type.

	Terminal [ms]	**User Device [ms]**	**Tag [ms]**
Elliptic Curve Type A			
Exponentiation	10	67	81
Pairing	15	125	-
Verification	192	-	-
Tag Proof Generation	-	-	444
User Proof Generation	-	448	-
Elliptic Curve Type D			
Exponentiation	4	38	40
Pairing	31	1050	-
Verification	271	-	-
Tag Proof Generation	-	-	277
User Proof Generation	-	273	-

scenarios and for different types of devices. Therefore, we provide the results of each protocol using one RFID tag. Furthermore, we present the crucial EC operations' benchmarks on a wide range of devices in Table 4.3. The time is measured in milliseconds and the values are an average of 10 measurements, as in the previous

Tab. 4.3: Benchmark results of all tested devices.

	Smart Card [ms]	**Phone** [ms]	**Watch** [ms]	**Raspberry Pi 3** [ms]	**PC** [ms]
Exponentiation	40	38	207	3.3	0.4
Pairing	-	1050	6571	31	2.4
Tag Proof Generation	277	154	900	18	4
User Proof Generation	441	273	1502	24	5
Verification	-	-	-	271	21

case. All measurements were performed by using the elliptic curve d159 from the PBC library. We did not consider Android devices as a terminal device, since the pairing operation requires too much time and therefore it is not usable in practice.

Figure 4.6 depicts the time required for a proof construction on different devices (MultOS smart card, Android smart phone and smart watch for various number of tags). These devices act as a user device.

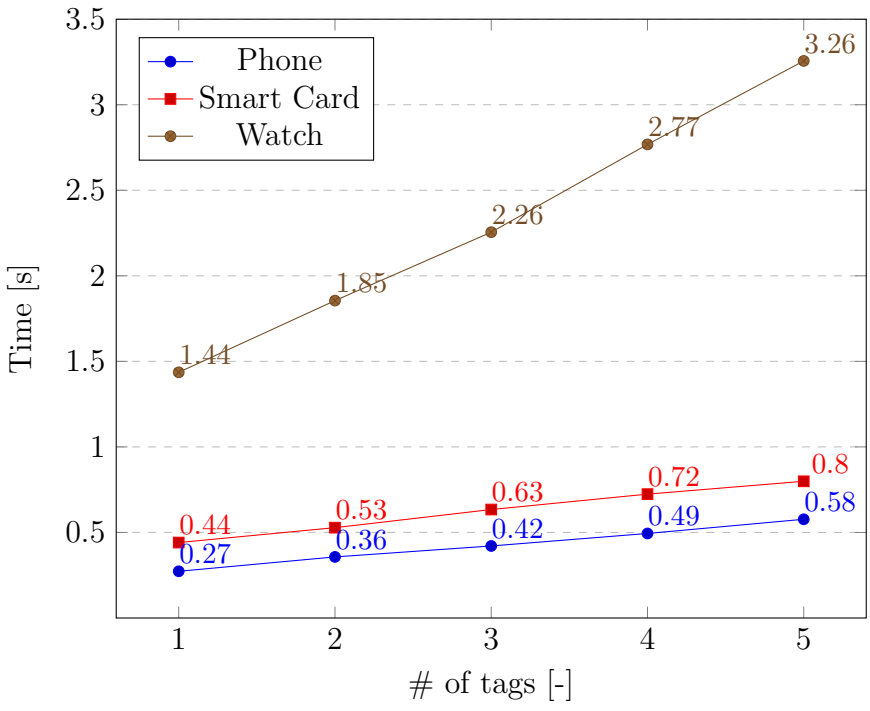

Fig. 4.6: Time dependence of the proof generation on the number of user device.

4.5.2 Revocation and Identification

Besides strong privacy-enhancing features, there must be also mechanisms to revoke and/or identify malicious users. All users are theoretically identifiable and traceable by their user IDs. However, these IDs are "hidden" in the signatures as the exponents. Due to the discrete logarithm problem assumption, one cannot easily get the identifiers and do the revocation and identification. However, our scheme is compatible with the major revocation schemes that are already available for cryptographic anonymous credential schemes [121, 120, 66]. In these revocation schemes, the hidden exponent (the user ID) is used as a revocation handle and can be disclosed only by designated authorities. Additionally, valid users remain anonymous while malicious users are identifiable and traceable by a designated authority, such as a court. Such schemes are provably secure, efficient and compatible without any modification, thus we refer to their specification (e.g., the scheme designed directly for smart cards [66]) in case revocation is needed.

4.6 Conclusion

We presented a cryptographic scheme that allows a novel approach for controlling physical access. Instead of the verification of fixed user or device identifiers, the terminals can check only the knowledge of such identifiers in a private manner, without explicitly exposing any personal information or the identifiers themselves. Furthermore, the presence of other RFID devices, possibly the safety equipment, can be enforced. Our protocols are based on proven cryptographic algorithms and are very practical - the proofs can be generated in under 500 ms on constrained devices, such as smart cards. We provided the full cryptographic description of all algorithms, the security and efficiency analysis and the implementation results on constrained devices. We find the scheme especially useful in applications where the physical access to dangerous environment is granted upon proving the presence of required safety equipment and where the strong privacy-protection regulation is enforced by law.

As for the future work, we will focus on the optimization of the verification algorithm, since the current verification time grows linearly with the number of tags involved in the authentication protocol. In particular, we would like to reduce the number of bilinear pairings which is the most time-consuming operation in the protocol.

5 Anonymous Data Collection Scheme from Short Group Signatures

The content of this chapter have been published in conference paper [14].

5.1 Introduction

Currently, there are proven cryptographic mechanisms that are able to guarantee the basic security properties in classical computer networks containing mostly PCs and servers. However, the structure of communication networks is changing in recent years and the infrastructures are becoming more and more heterogeneous, comprising industrial devices, small personal wearable devices, sensors, microcontrolers, etc. These devices are often very computationally constrained, which prevents the usage of standard cryptographic techniques for securing the communication. Lightweight cryptography mechanisms are being sought for the deployment on such devices. On the other side, the number of such personal devices is huge and quickly rising with the expansion of IoT networks, smart grids, cyber physical systems, etc. In some scenarios, millions of constrained devices communicate with one another and with central nodes. That is the case of sensor networks, in particular smart metering systems. In such applications, millions of relatively simple devices produce data that are collected by central nodes. Providing security in such environment is difficult, as the constrained devices are very limited in computational power and memory on one side and the central nodes must securely collect a very high number of messages from various sources on the other side.

In addition to the traditional requirements on confidentiality and authenticity, new demands on privacy protection are being imposed, mostly by EU regulations, but also by some US strategic plans [7]. That leads to the design of technologies that limit any disclosure of private information that is not necessary for the system functionality. However, such privacy-enhancing features are often extremely costly regarding computational resources.

In this chapter, we propose a cryptographic scheme based on group signatures that is designed to address the challenges identified above. Our anonymous data collection scheme allows constrained devices to efficiently generate group signatures on their data. Therefore, the collectors can be assured that data are collected from trusted sources and were not modified during a transfer. The signatures are fully anonymous, untraceable and unlinkable, thus supporting the full set of privacy-enhancing features. The collector learns that the signature was created by a trusted group member, but the concrete identity stays undisclosed. As an additional key

feature, our scheme also provides efficient revocation, i.e., a practical mechanism to identify and invalidate malicious users.

We present the full cryptographic description of all protocols of the scheme, show the efficiency analysis and the results of our implementation on devices with diverse computational power, from smart cards, microcontrollers to standard PCs. By showing also practical results, we prove the readiness of the scheme for a real-world use. Besides the straightforward use in data collection systems, we also note other applications, such as e-ticketing, transportation and e-IDs.

5.2 Related Work

The work on anonymous data collection schemes became intensive only very recently, with the deployment of smart metering technologies into practical installations. However, the cryptographic primitives, that are the main building blocks of these schemes, are known for more than a decade. The core building blocks are the group signatures, allowing users to create signatures using their private keys and verifiers to verify the signatures using a common public key. The research into group signatures was started by the seminal work of Chaum and van Heyst in [70]. A large number of group signatures has been proposed, e.g., in [92, 52, 78, 79, 122, 123, 124]. In particular, the scheme called BBS [52] serves as a fundamental building block for many security solutions (e.g., [125] and [126]). Recently, short randomizable signatures were proposed [127] that allow efficient proofs of signature knowledge and creating group signatures by signing the individual users' private keys by the manager's private key. For the verification, only the manager's public key is necessary. We take the same approach in our scheme.

Furthermore, some privacy preserving solutions based on pseudonyms have been proposed, e.g., in [128, 129, 130, 129, 130]. However, the solutions based on pseudonyms are usually inefficient as they require users to switch between many (pseudo)identities, thus need extensive cryptographic material and multiple keys.

Both the group signature schemes and the pseudonymous schemes mostly lack efficient revocation mechanisms that are scalable enough for large applications with millions of constrained user devices. Either the revocation function needs very expensive computations (such as bilinear pairings) or is rather tailored for authentication and access control schemes (such as [66, 88]).

As a result, we lack a practical data collection scheme that is provably secure, with short and fast signatures, with efficient revocation mechanisms and providing all privacy-enhancing features.

Our Contribution

We propose a novel cryptographic scheme that we call an anonymous data collection scheme that is instantiated using the wBB signature [25] and the efficient proofs of their knowledge [66]. On a general level, we take the approach of [127], i.e., we let the manager sign all users' private keys. The users then prove the knowledge of such a signature and verifier checks the proof using the manager's public key. Our scheme is unique in the following properties:

- provides all *privacy-enhancing* features: anonymity, unlinkability, untraceability,
- the signatures are *small* and *constant*: the size is below 169 B using a strong 224 b curve,
- the signature generation is *fast*: requires no bilinear pairing and only 5 exponentiations,
- the signature verification including *revocation* check is efficient: requires only 2 pairings and $\mathcal{O}(|RL|^1)$ exponentiations,
- the scheme is built using primitives with formal security proofs.

Besides the cryptographic design, we also provide the complete implementation results and benchmarks on a wide spectrum of devices, i.e. smart cards, microcontrollers and PCs. The practical results certify the usability in practice using contemporary cryptographic parameters recommended by NIST [2].

5.3 Cryptographic Design

Group signatures [70] allow users to sign messages using their private keys without being identifiable or traceable, as the signatures are verified using a single, general public key. There are many proposals for group signatures, focusing on size, speed of construction, security or advanced features. For our data collection scheme, we adopt the approach used in [127], i.e., we let the manager to sign the private keys of users (sk_i) using a signature scheme σ (wBB in our case) that allows efficient randomizable proofs of the signature knowledge, resulting in signature $\sigma(sk_i)$. Proving the manager's signature knowledge using signature proof of knowledge then allows the construction of user's group signatures on messages ($SPK\{sk_i : \sigma(sk_i)\}(m)$). Using such a construction, each user has his own private key but the signatures are verified using a common manager's key. Furthermore, all signatures are anonymous, untraceable and unlinkable.

[1] Revocation List

5.3.1 Requirements

The scheme algorithms must fulfil the following security and privacy properties of the group signatures defined by Bellare et al. [131]. The manager is trusted not to impersonate signers.

Security Requirements

- *Correctness*: signatures are verified correctly if and only if they are generated by valid and honest users.
- *Soundness*: invalid signatures generated outside the group must be rejected by the `Verify` protocol.

Privacy Requirements

- *Anonymity*: all signatures are anonymous, untraceable and unlinkable to all entities except the manager.
- *Traceability*: the manager can de-anonymize, link and trace signatures.

5.3.2 General Architecture

Three types of entities interact in our data collection scheme: a manager, a user and a collector.

- **Manager**: the manager generates cryptographic parameters and keys. It also enrols new users (devices) and revokes invalid ones.
- **User**: the user is represented by its device, such as a smart meter, sensor or some wearable device. It is the source of data that are signed and transferred to the central device (collector).
- **Collector**: the collector represents the central node that collects all data from users and verifies the group signatures.

The privacy-enhanced data collection scheme is presented in Figure 5.1. The entities interact in the following cryptographic algorithms and protocols.

$(pk, sk_m, par) \leftarrow$ `Setup`(1^κ): on the input of security parameter κ, the algorithm generates the public systems parameters par (implicit input of all other algorithms), the public key shared by all users pk and the private key of the manager sk_m. The `Setup` algorithm is run by the manager.

$(sk_i, rd) \leftarrow$ `Register`(id_i, sk_m): on the input of the manager's private key sk_m and the user identifier id_i, the register protocol outputs the user's private key sk_i and updates the manager's revocation database rd. The `Register` algorithm is run as an interactive protocol between the manager and the user.

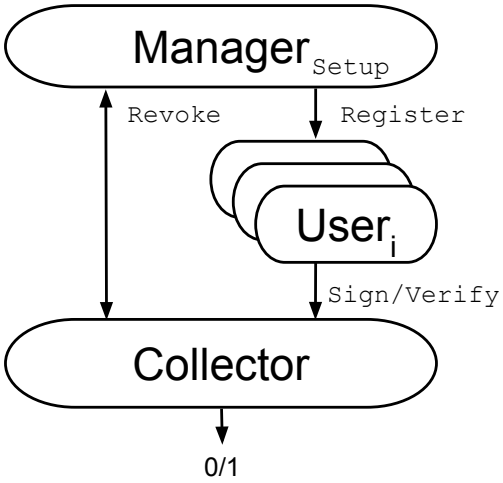

Fig. 5.1: Architecture of the scheme proposed.

$sig(sk_i, m) \leftarrow \texttt{Sign}(m, id_i, sk_i)$: on the input of the message m, user's identifier id_i and its private key sk_i, the algorithm outputs the signature on the message $sig(sk_i, m)$. The algorithm is run by the user.

$(0/1) \leftarrow \texttt{Verify}(sig(sk_i, m), m, pk)$: on the input of the message m, signature $sig(sk_i, m)$ and the public key pk, the algorithm returns 1 iff the signature is valid and 0 otherwise. The algorithm is run by the collector.

$id_i \leftarrow \texttt{Revoke}(rd, sig(sk_i, m))$: on the input of the manager's revocation database rd and a signature $sig(sk_i, m)$, the algorithm outputs the identifier id_i of the signer.

5.3.3 Cryptography Specification

We instantiate the algorithms of the data collection scheme presented in the previous section using the wBB signature [25] and its efficient proof of knowledge [66]. On a high level, we let the user to obtain a wBB signature on his private identifier from the manager. Then, the user proves the knowledge of such a signature anonymously and efficiently using the Schnorr-like zero-knowledge protocol for proving the knowledge of a discrete logarithm [17]. For the conversion from the proof of knowledge to the signature, we use the Fiat-Shamir heuristics [64]. We present the concrete algorithm and protocol instantiations below.

`Setup`

$(pk, sk_m, par) \leftarrow$ `Setup`(1^κ): the algorithm inputs the security parameter κ and generates the bilinear group with parameters $par = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g \in \mathbb{G}_1, g_2 \in \mathbb{G}_2)$ satisfying $|q| = \kappa$. It also generates the manager's private key $sk_m \xleftarrow{\$} \mathbb{Z}_q$ and computes the public key $pk = g_2^{sk_m}$. It outputs the (pk, par) as a public output and the sk_m as the manager's private output.

`Register`

$(sk_i, rd) \leftarrow$ `Register`(id_i, sk_m): the protocol is distributed between the user and the manager. The manager inputs his private key sk_m and the user inputs his private identifier id_i. The protocol outputs the wBB signature $sk_i = g^{\frac{1}{sk_m + id_i}}$ to the user and updates the manager's revocation database rd by storing id_i.

`Sign`

$sig(sk_i, m) \leftarrow$ `Sign`(m, id_i, sk_i): the algorithm inputs the user's private identifier id_i, his private key sk_i and the message to be signed. It outputs the signature $sig(sk_i, m)$ that consists of the following elements $(g', sk_i', \bar{sk_i}, \pi)$:

- $g' = g^r$: the generator raised to a randomly chosen randomizer $r \xleftarrow{\$} \mathbb{Z}_q$.
- $sk_i' = sk_i^r$: the users's private key raised to the randomizer.
- $\bar{sk_i} = sk_i'^{-id_i}$: the randomized private key raised to the user identifier.
- $\pi = SPK\{(id_i, r) : \bar{sk_i} = sk_i'^{-id_i} \wedge g' = g^r\}(m)$: proof of knowledge of r and id_i signing the message m.

`Verify`

$(0/1) \leftarrow$ `Verify`$(sig(sk_i, m), m, pk, bl)$: the algorithm inputs the massage m, its signature $(sig(sk_i, m)$, a blacklist bl and the public key pk. It checks the proof of knowledge signature π and checks that the signature is valid with respect to the manager's public key using the equation $\mathbf{e}(\bar{sk_i} g', g_2) \stackrel{?}{=} \mathbf{e}(sk_i', pk)$. The collector also performs the revocation check $\bar{sk_i} \stackrel{?}{=} sk_i'^{-id_i}$ for all id_i values stored on the blacklist bl. If the revocation check equation holds for any value on the blacklist, the signature is rejected. Otherwise, the signature is accepted if all other checks pass.

`Revoke`

$bl \leftarrow$ `Revoke`$(rd, sig(sk_i, m))$: the algorithm inputs a signature $sig(sk_i, m)$ and a revocation database rd. It checks $\bar{sk_i} \stackrel{?}{=} sk_i'^{-id_i}$ for all id_is in rd. The id_i that holds in the equation is put on a public blacklist bl.

The `Register`, `Sign` and `Verify` algorithms are presented in CS notation in Figure 5.2.

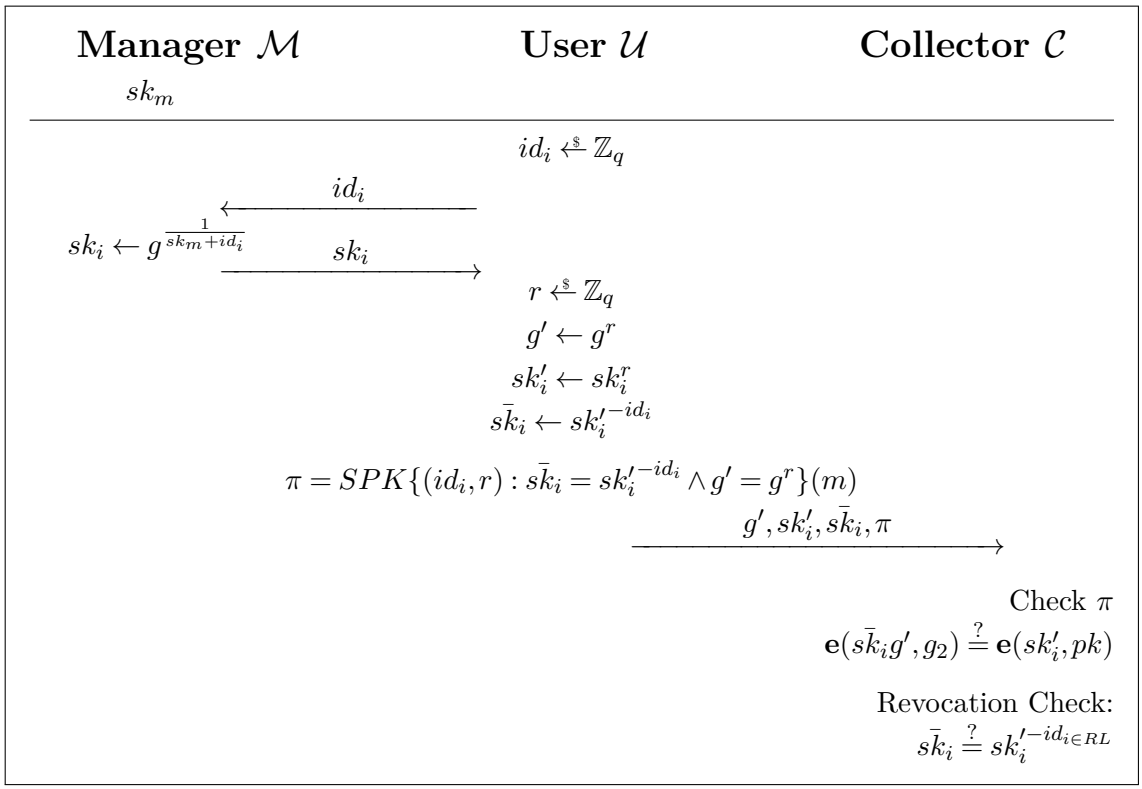

Fig. 5.2: `Register`, `Sign` and `Verify` algorithms.

5.4 Security Analysis

We imply the security of our scheme from the security of the building blocks. The wBB signature scheme used for signing the private keys is unforgeable against a non-adaptive chosen message attack under the q-SDH assumption [25]. The group signature is an efficient proof of knowledge based on the standard zero-knowledge proofs [17] that are *complete*, *sound* and *zero-knowledge* in the random oracle model.

Proof. Completeness: honest users pass the collector's check.

The proof π is always accepted for valid signatures, due to the completeness of the proof of knowledge protocol. The proof π is always reject for the invalid signatures, due to the soundness property of the proof of knowledge protocol. The pairings are always accepted if a valid manager's key is used in the signature:

$$\mathbf{e}(\bar{sk_i}g', g_2) = \mathbf{e}(sk'_i, pk) \tag{5.1}$$
$$\mathbf{e}(sk_i^{-id_i r} g^r, g_2) = \mathbf{e}(sk_i^r, g_2^{sk_m}) \tag{5.2}$$
$$\mathbf{e}(g^{\frac{-id_i r}{sk_m + id_i}} g^r, g_2) = \mathbf{e}(sk_i^r, g_2^{sk_m}) \tag{5.3}$$
$$\mathbf{e}(g^{\frac{sk_m r + id_i r - id_i r}{sk_m + id_i}}, g_2) = \mathbf{e}(sk_i^r, g_2^{sk_m}) \tag{5.4}$$
$$\mathbf{e}(sk_i^{sk_m r}, g_2) = \mathbf{e}(sk_i^r, g_2^{sk_m}) \tag{5.5}$$
$$\mathbf{e}(sk_i, g_2)^{sk_m r} = \mathbf{e}(sk_i, g_2)^{sk_m r} \tag{5.6}$$

□

Proof. Soundness: only registered users pass collector's check.

Assume a user who is not registered (i.e., does not know the identifier id_i) and passes the collector's check for two different challenges e and e' with two different responses s and s':

$$\widehat{t} = (\bar{sk_i}g')^e sk_i'^{s_{id_i}} g^{s_r} \tag{5.7}$$
$$\widehat{t} = (\bar{sk_i}g')^{e'} sk_i'^{s'_{id_i}} g^{s'_r} \tag{5.8}$$

and we get:

$$(\bar{sk_i}g')^{e'-e} = sk_i'^{s_{id_i} - s'_{id_i}} g^{s_r - s'_r} \tag{5.9}$$

and therefore:

$$(\bar{sk_i}g') = sk_i'^{\frac{s_{id_i} - s'_{id_i}}{e'-e}} g^{\frac{s_r - s'_r}{e'-e}} \tag{5.10}$$

Thus the user can efficiently compute both the randomizers $r = \frac{s_r - s'_r}{e'-e}$ and the identifier $id_i = \frac{s_{id_i} - s'_{id_i}}{e'-e}$ and we reached the contradiction to our original assumption. □

Proof. Anonymity: The proof π is always anonymous, untraceabe and unlinkable due to the zero-knowledge property of the proof of knowledge protocol.

Distribution of $g', sk'_i, \bar{sk_i}$ is random and uniform in $\mathbb{Z}_q$ as r is selected randomly and uniformly from $\mathbb{Z}_q$. Thus, the values disclosed are indistinguishable from random elements. □

Proof. Traceability: Provided the user's private identifier id_i, the signatures are linkable by $\bar{sk}_i = sk_i'^{-id_i}$. □

5.5 Implementation and Performance Analysis

We present the computational and communication complexity analysis in this Section. Furthermore, we present the results of our practical implementation on several types of devices, including constrained devices and wearables.

The user has to compute only 5 exponentiations to construct the proof. On the other side, the verifier has to perform 2 bilinear pairings and 3 exponentiations to verify the proof. The revocation check time is linear to the number of revoked users and, therefore, requires $\mathcal{O}(|RL|)$ exponentiations, where $|RL|$ is the number of revoked users. The computational and communication costs of our scheme is considerably reduced due to the use of EC cryptography which requires smaller keys compared to traditional protocols on a similar security level. Our signatures contain only 3 elements of $\mathbb{G}_1$, and 3 elements of $\mathbb{Z}_p$. Therefore, using a strong 224 b elliptic curve, only 255 B need to be sent as a signature. In case that EC point compression is used, we can reduce the signature size to less than 169 B (1347 b). Hence, the size of $\mathbb{Z}_p$ remains 224 b and the size of each element of $\mathbb{G}_1$ is 225 b rather than 448 b. This is especially significant in smart card communication scenarios, where the payload size of APDU message is restricted to 255 B if T=0 transmission protocol is used.

We implemented the `Sign` and `Verify` protocols, the full description of our algorithms is in Figure 5.3.

5.5.1 Performance Analysis

Our proposal is particularly suitable for data collections systems, such as smart metering. In these systems, the data are anonymously collected by a central collector from the remote nodes. Furthermore, due to the fast signature generation speed and size efficiency, our scheme can by used in a wide range of other applications, such as e-ticketing and transportation eIDs. For this reason, we performed the measurements on different kinds of devices, both constrained (wearables, embedded devices) and powerful (PC, server) ones. We considered the following test scenarios:

Smart metering: smart houses are equipped with different types of sensors, e.g., for gas, water, or electricity consumption detection. The collected data are sent to an energy supplier (collector) who performs statistical evaluations on the consumption in a given area. The consumption profile of a concrete user must remain anonymous, thus the application cannot be used directly for billing purposes. However, if

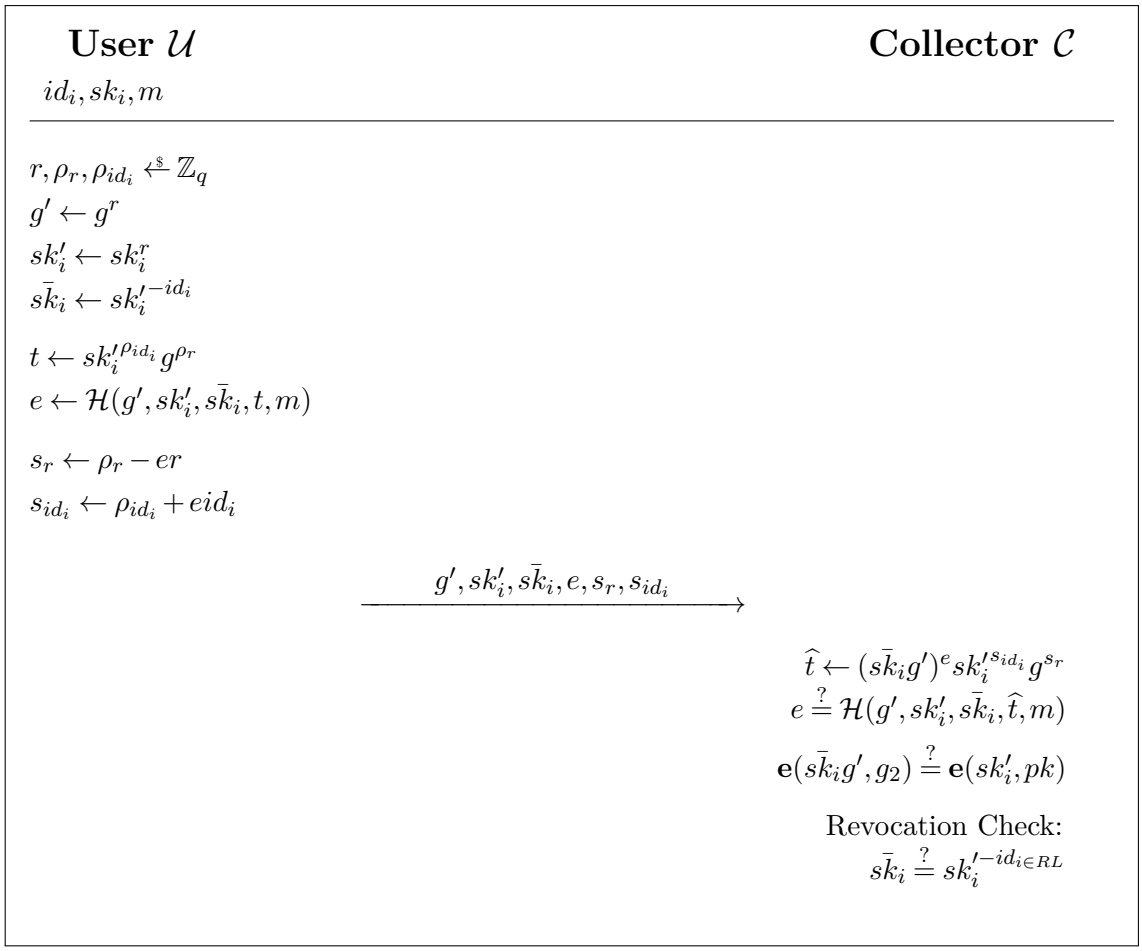

Fig. 5.3: Implementation of `Sign` and `Verify` algorithms.

a non-standard household consumption is detected, the energy supplier can request the identity of the "malicious" user from a trusted third party. In this scenario, the smart house sensors can be represented Raspberry Pi devices, while PC and server can act as a collector.

E-ticketing: wearable devices, such as smart cards, smart phones and smart watch, can be used for storing tickets. Validity of a ticket can be checked by a terminal, e.g. installed in a vehicle. In this scenario, Raspberry Pi device represents a terminal and a wearable acts as a user's device. The verification can be performed locally on the terminal or remotely on the powerful central server.

We performed the measurement on all devices mentioned above. The detailed hardware and software specifications are described in Table 5.1.

Tab. 5.1: Specification of tested devices.

Device	**CPU/MCU**	**OS**	**RAM**
Smart Card	SC23Z018	MultOS 4.3.1	1.75 KB
Phone 1	Kirin 655	Android 7.0	3 GB
Phone 2	Krait 400	Android 5.1	2 GB
Watch 1	ARM Cortex-A7	Android 6.0	512 MB
Watch 2	ARM Cortex-A7	Android 7.0	768 MB
Raspberry Pi 3	ARM Cortex-A53	Raspbian 9.3	1 GB
Raspberry Pi 2	ARM Cortex-A7	Raspbian 9.3	1 GB
Raspberry Pi	ARM1176JZF-S	Raspbian 9.3	512 MB
PC	Intel i7-7700	Debian 8.6	16 GB
Server	Intel Xeon 2.27	Debian 8.6	32 GB

Note: Smart Card – ML4, Phone 1 – HUAWEI P9 Lite 2017, Phone 2 – SONY Experia Z1 Compact, Raspberry Pi 3 – Raspberry Pi 3 Model B, Raspberry Pi 2 – Raspberry Pi 2 Model B, Raspberry Pi – Raspberry Pi Model B+, Watch 1 - Sony SmartWatch 3 SWR50, Watch 2 – HUAWEI Watch 2

The performance tests required the implementation of the proposed scheme on different platforms and operation systems. In case of the smart card application, only standard MultOS API and free public development environment (Eclipse IDE for C/C++ Developers, SmartDeck 3.0.1, MUtil 2.8) were used. The application is written in MULTOS assembly code and C language. Smart phones and smart watches run an Android application written in Java language. In particular, we used Android Studio 3.0.1 as the official IDE for Android app development along with Android SDK depending on the specific device, and jPBC-2.0.0 [84] library which allows performing operations over elliptic curves (point addition, scalar multiplication and bilinear pairing). The rest of the devices run OS Linux and, therefore, the scheme was implemented in C, where PBC-0.5.14 [56] library was used for the elliptic curve operations. The scheme was developed in NetBeans IDE 8.2 development environment. The code was remotely build and executed on the targeted device, i.e., Raspberry Pi/2/3, PC and server.

The `Sign` and `Verify` algorithms were implemented using pairing-friendly elliptic curves. Since our scheme requires asymmetric bilinear pairing, we considered the elliptic curves of D types from the PBC library, namely d159, d201, and d224. The performance tests were run 10 times on each device, and the arithmetic mean of the measured values was calculated. The computation time of `Sign` and `Verify` algorithms is provided in Table 5.2. At the first sight, the effectiveness of `Sign` protocol is obvious. Using the 224 b elliptic curve, which is of 112 b security strength,

the `Sign` protocol takes only 442 ms on a smart card. On the other hand, the Android devices are slow in EC operations, in particular in bilinear pairing. In fact, Table 5.3, which provides the benchmarks of the crucial elliptic curve primitive operations on the tested devices, shows that Watch 1 and Watch 2 are slower than smart cards although they are much more powerful. This is due to the use of the jPBC library, which is a library written in Java rather than in C. Furthermore, hardware acceleration of EC operations is employed on smart cards.

Tab. 5.2: Performance of `Sign` and `Verify` protocols for different elliptic curves on various user devices.

	Signing time [ms]			**Verification time [s]**		
Device/Curve	**d159**	**d201**	**d224**	**d159**	**d201**	**d224**
Smart Card	362	415	442	–	–	–
Phone 1	180	253	336	2.1	2.5	3.1
Phone 2	665	705	943	10.9	11.6	12.7
Watch 1	1252	2215	2889	26.2	31.0	38.0
Watch 2	1019	1139	1637	13.6	15.8	19.2
Raspberry Pi 3	18	24	30	0.082	0.115	0.138
Raspberry Pi 2	32	42	53	0.144	0.197	0.236
Raspberry Pi	67	89	110	0.266	0.372	0.434
PC	3	4	5	0.007	0.009	0.011

The Figure 5.4 and Figure 5.5 show the time needed to complete the malicious user identification and revocation check procedure. The user identification procedure requires to perform scalar multiplications on the considered device (the cost of this operation is depicted in Table 5.3). In case of the de-anonymisation procedure, the number of scalar multiplications is equal to the number of users. We stress, that the de-anonymisation procedure is expected to be performed on powerful devices and can be parallelized on their processors and cores (CPU/Cores). For instance, our PC (1/4), and server (2/8) are able go through the list of thousands of users and find the identity of a user in less than 4 min, see Figure 5.4.

In the revocation check procedure, the PC (1/4) and server (2/8) are able to search the blacklist in less than 0.5 s, see Figure 5.5.

5.5.2 Complexity Analysis

In this section, we provide the comparison of our scheme with the state of the art group signature schemes. We considered the efficient group signature schemes identified in [77]. Table 5.4 shows the comparison of our scheme with these pairing and

Tab. 5.3: Benchmarks of primitive operations.

Curve	**d159**		**d201**		**d224**	
EC operation	$E_{\mathbb{G}_1}$ [ms]	**e** [s]	$E_{\mathbb{G}_1}$ [ms]	**e** [s]	$E_{\mathbb{G}_1}$ [ms]	**e** [s]
Smart Card	40	–	44	–	50	–
Phone 1	38	1.0	48	1.2	65	1.4
Phone 2	153	5.4	161	5.7	187	6.7
Watch 1	350	12.4	457	14.7	548	18.5
Watch 2	196	6.5	246	7.5	325	9.1
EC operation	$E_{\mathbb{G}_1}$ [ms]	**e** [ms]	$E_{\mathbb{G}_1}$ [ms]	**e** [ms]	$E_{\mathbb{G}_1}$ [ms]	**e** [ms]
Raspberry Pi 3	3.3	31.6	4.7	45.3	5.8	55.2
Raspberry Pi 2	6.0	54.8	7.9	77.4	10.2	94.5
Raspberry Pi	12.8	97.9	17.2	140.1	21.1	167.6
PC	0.4	2.1	0.5	2.9	0.7	3.6
Server	0.2	1.9	–	–	0.3	3.3

Note: $E_{\mathbb{G}_1}$ – EC scalar multiplication in G_1, **e** – bilinear paring, $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.

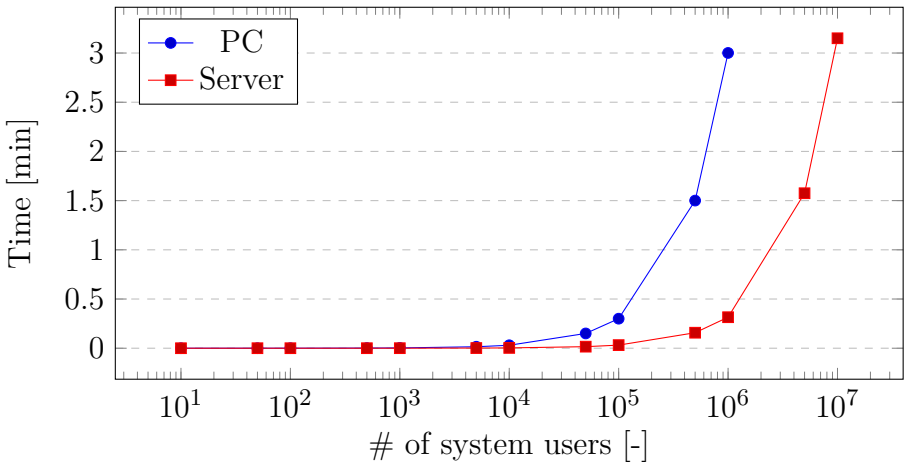

Fig. 5.4: Time needed to identify a malicious user.

non-pairing based group signature schemes. Bilinear pairing and group exponentiation operations are denoted as **e** and $E_{\mathbb{G}}$, respectively. The execution time of each operation depends on the bitlength of the elements in respective groups and

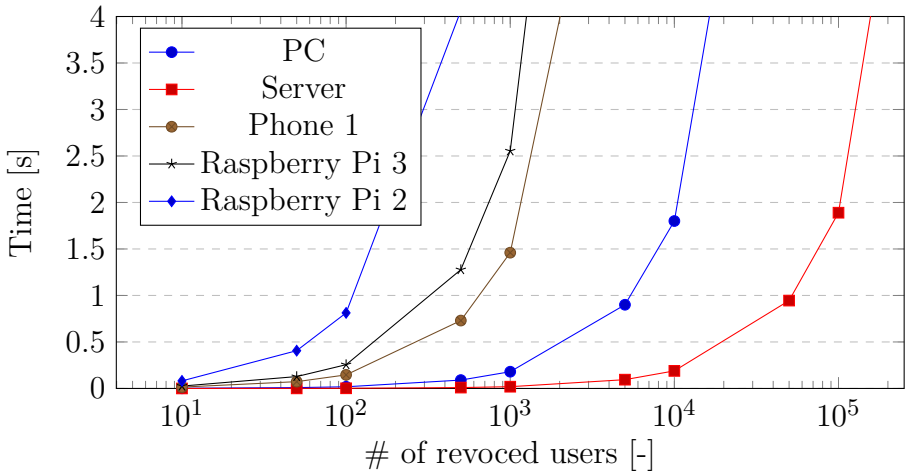

Fig. 5.5: Time needed to check the black list.

fields. In the pairing-based schemes, $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, $\mathbb{Z}_p$ denote different groups with the following bitlengths: $|\mathbb{G}_1| = 175$ b, $|\mathbb{G}_2| = 175$ b and $|\mathbb{G}_T| = 1050$ b computed as $k \cdot |\mathbb{G}_1|$, where k is the embedding degree (e.g. $k = 6$). $|\mathbb{Z}_p| = 170$ b denotes the field size of an elliptic curve. In the non-pairing schemes, $|\mathbb{G}_n^*| = 1024$ b denotes the multiplicative RSA group with exponents from $|\mathbb{Z}_q| = 160$ b. The total length of signatures depends on the security level chosen.

Tab. 5.4: Comparison with current short group signature schemes.

Scheme	Sign Cost	Verify Cost	Signature Size
BBS [52]	$9E_{\mathbb{G}_1}+3E_{\mathbb{G}_T}$	$1\mathbf{e}+8E_{\mathbb{G}_1}+2E_{\mathbb{G}_2}+3E_{\mathbb{G}_T}$	$3\mathbb{G}_1+6\mathbb{Z}_p$ (1545 b)
DP [78]	$8E_{\mathbb{G}_1}+3E_{\mathbb{G}_T}$	$1\mathbf{e}+7E_{\mathbb{G}_1}+2E_{\mathbb{G}_2}+3E_{\mathbb{G}_T}$	$4\mathbb{G}_1+5\mathbb{Z}_p$ (1559 b)
HLCCN [79]	$7E_{\mathbb{G}_1}+5E_{\mathbb{G}_T}$	$1\mathbf{e}+5E_{\mathbb{G}_1}+2E_{\mathbb{G}_2}+4E_{\mathbb{G}_T}$	$3\mathbb{G}_1+5\mathbb{Z}_p$ (1375 b)
ACJT [74]	$12E_{\mathbb{G}_n^*}$	$10E_{\mathbb{G}_n^*}$	$7\mathbb{G}_n^*+1\mathbb{Z}_c$ (7328 b)
CG [80]	$10E_{\mathbb{G}_n^*}$	$10E_{\mathbb{G}_n^*}$	$8\mathbb{G}_n^*+1\mathbb{Z}_q$ (8352 b)
IMSTY [81]	$7E_{\mathbb{G}_n^*}$	$7E_{\mathbb{G}_n^*}$	$5\mathbb{G}_n^*+5\mathbb{Z}_p+1\mathbb{Z}_c$ (6155 b)
HM GS [82]	$9E_{\mathbb{G}_n^*}$	$10E_{\mathbb{G}_n^*}$	$7\mathbb{G}_n^*+1\mathbb{Z}_q$ (7328 b)
Our Scheme	$5E_{\mathbb{G}_1}$	$2\mathbf{e}+3E_{\mathbb{G}_1}$	$3\mathbb{G}_1+3\mathbb{Z}_p$ (1035 b)

Note: $E_{\mathbb{G}_1}$ – EC scalar multiplication in $\mathbb{G}_1$, similarly $E_{\mathbb{G}_2}$ and $E_{\mathbb{G}_T}$, $\mathbf{e}$ – bilinear pairing.

5.6 Conclusion

We presented a novel data collection scheme which is more efficient than comparable state of the art schemes as shown in the comparative complexity analysis. The proposed scheme is built using primitives with formal security proofs and the security of the proposed scheme itself was proven.

Our proposal is particularly suitable for data collections systems, such as smart metering. However, our scheme can be also used in other areas of IoT, such as smart grids, Industry 4.0, e-ticketing, transportation eIDs, due to the signature generation speed and short size.

Moreover, we provided the full implementation results from a wide range of devices, including IoT devices, to show the efficiency of our solution. A signature on the 112 b security level can be generated in 442 ms on a standard smart card, in 336 ms on a current smart phone and in 18 ms on the Raspberry Pi 3. Furthermore, our scheme provides fast revocation checks, the blacklisted user can be identified in less than 2 s.

6 Anonymous Credentials with Practical Revocation

The content of this chapter have been published in conference paper [15].

6.1 Introduction

Current authentication schemes use identity-based authentication approach, i.e., a user reveals his identity to a verifier and then proves the identity ownership. In other words, a verifier asks a user the question "*Who are you?*" at first. This question has a big impact on user's privacy. Nevertheless, user's identity does not need to be always disclosed. For instance, a hospital wants to give an access to the Human Immunodeficiency Virus (HIV) discussion group to a user who presents this disease, but a patient could be reluctant to participate in the group if he has to reveal his identity. In many cases, it is enough to know only some particular user's attributes.

Attribute-Based Credential schemes are modern cryptographic schemes which provide higher protection of users' privacy during a verification phase. Users can anonymously prove the possession of some personal attributes without disclosing their identity or any other sensitive information. ABC schemes change the question from "*Who are you?*" to "*What can you do?*", which is more privacy-preserving for a user and sufficient for a verifier. A user holds some personal attributes, and during the verification phase, he proves the possession of the attributes required by a verifier. The examples of attributes include age, citizenship, gender or nationality for eIDs, validity of ticket for public transportation, etc.

Elliptic curves cryptography provides security level comparable to classic systems while using fewer bits and less computing power. For this reason, elliptic curve cryptography is very suitable for ultra-low-power devices, such as smart cards. Nowadays, there are only few well-known ABC schemes such as U-Prove [87], Idemix [21], and HM12 [88], for more details see Chapter 3.2. Most of these schemes can be easily constructed over elliptic curves, however, except the HM12 scheme.

6.2 Related Work

Current anonymous credential schemes allow users to prove the possession of their attributes without disclosing user's (attribute holder's) identity. Furthermore, these schemes also provide untraceability, which means that an issuer, who issued attributes to a user, is not able to track the user during the verification phase. U-prove is a cryptographic technology maintained by Microsoft Corporation. The main

drawback of the scheme is the session linkability, i.e., all anonymous credentials of a single user are mutually linkable. Moreover, U-prove does not provide features for malicious user identification. Idemix technology (Identity Mixer) is an anonymous credential system developed at IBM Research in Zurich. Idemix provides session unlinkability. On the other hand, there is no universal efficient revocation mechanism, therefore it is not possible to directly revoke users' credentials and identify malicious users. At last, HM12 scheme solves all drawbacks of the previous schemes by providing anonymity, untraceability, unlinkability, selective disclosure of attributes and non-transferability. Revocation of anonymous credentials, their unlinkability and identification of malicious users are made possible by using Okamoto-Uchiyama trapdoor [67]. The main drawback of the scheme is lower computation efficiency. The `Prove_Att` phase is significantly slower compared to U-Prove and Idemix schemes if same security strength held. Moreover, the `Prove_Att` phase complexity grows linearly on number of blacklisted users. On the other hand, the needed time for `Prove_Att` phase does not depend on number of hidden attributes but only on disclosed ones.

The most efficient implementation of the U-prove protocol was done on a MultOS card and was described in [91], and it is depicted in Figure 3.4. The proof of attribute ownership is faster than 1 s if 5 attributes are issued. Idemix was also implemented on MultOS card and its proof of attribute ownership needs less than 1.5 s if 5 attributes are issued, see Figure 3.7. The HM12 proof-of-concept implementation on MultOS ML2-80K-65 was provided in the original paper [88]. The authors assumed needed `Prove_Att` time around 2 s. The final optimized implementation on MultOS ML3 was provided [96] with total `Prove_Att` time ca. 2.9 s. Our implementation on the newest MultOS ML4 smart card that was developed within this thesis requires only ca. 1.6 s for one attribute disclosing, see Figure 3.10.

Our Contribution

In this Chapter, we present a novel elliptic curve variant of the HM12 attribute-based credential scheme [88]. The elliptic curve variant (we call it ecHM12) meets all requirements for an ABC scheme as well as the original scheme HM12. In particular, we preserved anonymity, untraceability, unlinkability, non-transferability, selective disclosure of attributes, computationally efficient revocation and malicious user identification. Furthermore, by involving elliptic curves to the scheme, we achieve higher computational efficiency compared with the standard HM12 scheme, especially during the `Prove_Att` phase. The ecHM12 scheme also requires smaller bandwidth, since data communication transfer is 85% smaller compared to the original scheme HM12.

6.3 Cryptographic Design

At first, the definition of requirements on our novel elliptic curve ABC scheme is listed. Then, we define the algorithms and entities in the scheme. At last, we present the concrete instantiation of the ABC scheme based on elliptic curve cryptography.

6.3.1 Requirements

We require the scheme to be secure and to preserve all privacy-enhancing properties from the original scheme HM12. Moreover, we require the scheme to be computationally more efficient and more bandwidth friendly compared to the original scheme. It is equally important to maintain revocation properties of the original scheme, i.e. the scheme provides immediate revocation, issuer and verifier driven revocation, the Verifier Local Revocation (VLR) and computationally efficient revocation. We can divide the requirements mentioned above as follows:

Security Requirements

- *Completeness*: if a User holds required attributes, he always pass the `Prove_Att` protocol.
- *Soundness*: a User cannot pass the `Prove_Att` protocol without possession of the attributes which are required by the Verifier.
- *Zero-Knowledge*: the `Prove_Att` protocol transcript must be simulatable without the knowledge of the attributes secret keys, thus provably release no sensitive information.
- *Non-transferability*: a User is equipped with a unique private key, which is stored on a secure element, e.g. a smart card.

Privacy Requirements

- *Anonymity*: each User anonymously proves possession of attributes. Therefore, his identity and his behaviour remains hidden in the system.
- *Untraceability*: all credentials are randomized, i.e., the Issuer is not able to track User's movements and behaviour.
- *Unlinkability*: all single user's sessions are mutually unlinkable. Therefore, the Verifier or an eavesdropper are not able to link individual sessions together and profile the User.
- *Selective disclosure of attributes*: a User can choose the attributes which have to be disclosed, other attributes remain hidden.

Efficiency Requirements

- *Speed*: the scheme must be practically usable in current card-based access control systems, therefore, the verification time should be around 1 s or less.
- *Readiness for smart card*: the scheme must be fast even on constrained devices, such as smart cards. No operations, that are unavailable on smart cards (such as bilinear pairings), can be used in User's algorithms.
- *Revocation*: the Manager is able to revoke a User or credentials or unlinkability property or even to disclose User's identity. All these properties are immediately feasible and invoked by the User, the Manager, the Issuer, or the Verifier.

6.3.2 General Architecture

Three types of entities interact in our ABC scheme:

- **User**: gets issued attributes from the Issuer and anonymously proves their possession to the Verifier.
- **Issuer**: is responsible for issuing user attributes.
- **Manager**: validates user credentials (collection of attributes issued by the Issuer), can revoke a (dishonest) User, and in collaboration with the Issuer, can identify the (dishonest) Users.
- **Verifier**: verifies possession of required attributes provided by Users.

Each entity communicates in the system through specific cryptographic protocols. All the protocols and involved entities are depicted in the Figure 6.1.

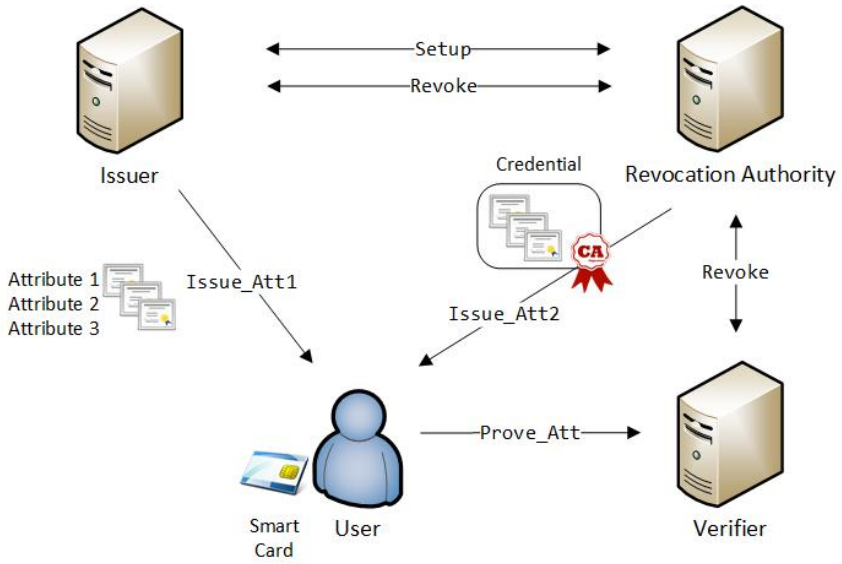

Fig. 6.1: Architecture of proposed ecHM12 scheme.

For a better understanding, the used protocols are shortly described below. The full specification is provided in the next section.

$(par, K_{\mathcal{M}}, K_{\mathcal{I}}) \leftarrow$ `Setup`(1^{κ}): the protocol is run between the Issuer and the Manager. The input parameter is the secure parameter κ and the output are public system parameters *par* that are constant in the system. Further, the Issuer and the Manager generate their $K_{\mathcal{I}}$ and $K_{\mathcal{M}}$ key pairs.

$(K_U) \leftarrow$ `Issue_Att`$(par, K_{\mathcal{M}}, K_{\mathcal{I}})$: the Issuer, the Manager and the User are involved in this protocol. The main purpose of this protocol is to compute User's key K_U, which permits user to prove the attributes possession within the `Prove_Att` protocol.

$(0/1) \leftarrow$ `Prove_Att`(par, K_U): the protocol is run between the User and the Verifier. The User proves the ownership of the attributes to the Verifier (the PK protocols are used for this purpose). The User's public output is the generated *proof* while the Verifier's public output is a decision on the *proof* acceptance (0/1).

$(BL) \leftarrow$ `Revoke`$(par, proof, K_{\mathcal{M}})$: the protocol is run between the Verifier, the Manager and eventually the Issuer (in case of anonymity revocation). It is launched in case that the User must be removed from the system due to various reasons (smart card lost, dishonest user or other justifiable reasons). The Verifier sends the *proof* generated by the User to the Manager. Then, the Manager is able to remove the User from the system by using the *proof* and the database of currently validated credentials.

6.3.3 Cryptography Specification

In this section we provide a detailed description of each protocol which runs within the proposed scheme.

`Setup`

$(par, K_{\mathcal{M}}, K_{\mathcal{I}}) \leftarrow$ `Setup`(1^{κ}): the `Setup` protocol mostly matches the original HM12 scheme, only in the final step the scheme is switched to the elliptic curve variant. The main purpose of this protocol is to establish system parameters *par* and to generate $K_{\mathcal{I}}$ and $K_{\mathcal{M}}$ keys. The input parameter κ defines the security strength of the cryptographic scheme, similarly to the scheme HM12. Additionally, κ includes elliptic curve domain parameters bitlengths. The Issuer defines a group $\mathbb{H}$ modulo big prime number p and generators h_1, h_2 of order q, with $q|p-1$. $\mathbb{H}$ is the subgroup

of the group $\mathbb{Z}_p^*$ as in the DSA signature scheme. In addition, the Issuer generates the key pair $sk_{\mathcal{I}} = K_{\mathcal{I}}$ and $pk_{\mathcal{I}}$ for signing purpose using a defined signature scheme, e.g. RSA. The Manager needs to:

- define the Okamoto-Uchiyama group $\mathbb{OU}_n$ by specifying the modulus $n = r^2 s$, where r and s are secure primes (i.e. $r = 2r' + 1$, $s = 2s' + 1$, where r' and s' are primes),
- find a generator $g_1 \xleftarrow{\$} \mathbb{Z}_n^*$ of $ord(g_1 \bmod r^2) = r(r-1)$ in $\mathbb{Z}_{r^2}^*$ and $ord(g_1) = rr's'$ in $\mathbb{Z}_n^*$,
- choose an elliptic curve over finite field $E(\mathbb{F}_p)$ with the domain parameters (a, b, p, q, G, h), where p is big prime number specifying the field $\mathbb{F}_p$, $a, b \in \mathbb{F}_p$ are static coefficients of the E, G is curve point generator $G = (x_G, y_G)$ of order q, and h is the cofactor defined as $h = \#E(\mathbb{F}_p)/q$,
- randomly choose Master's secrets $\langle s_{1,i} \rangle_{i \in \mathcal{A}} \xleftarrow{\$} \mathbb{Z}_q$ for available attributes, and $s_2, s_3 \xleftarrow{\$} \mathbb{Z}_q$, such that $\langle GCD(s_{1,i}, q) = 1 \rangle_{i \in \mathcal{A}}$, $GCD(s_2, q) = 1$, $GCD(s_3, q) = 1$.
- compute second generator $g_2 \leftarrow g_1^{s_2} \bmod n$ in the $\mathbb{OU}_n$,
- set first curve generator $G_1 \leftarrow G$, generate all attributes $\langle eca_i \leftarrow s_{1,i} \bullet G_1 \rangle_{i \in \mathcal{A}}$, and finally, get second and third curve generators $G_2 \leftarrow s_2 \bullet G_1$, $G_3 \leftarrow s_3 \bullet G_1$ in $E(\mathbb{F}_p)$.

The system parameters $par = (g_1, g_2, \mathbb{OU}_n, h_1, h_2, \mathbb{H}, G_1, G_2, G_3, E(\mathbb{F}_p))$ together with the set of attributes $\langle eca_i \rangle_{i \in \mathcal{A}}$ are made public, while the values r, s and $\langle s_{1,i} \rangle_{i \in \mathcal{A}}, s_2, s_3$ represent the Manager's secret key $K_{\mathcal{M}}$ and are securely stored by the Manager, and the secret key $K_{\mathcal{I}}$ is securely stored by the Issuer.

`Issue_Att`

$(K_U) \leftarrow$ `Issue_Att`$(par, K_{\mathcal{M}}, K_{\mathcal{I}})$: the protocol follows the HM12 idea. The issue phase is split into two parts `Issuer_Att1` and `Issuer_Att2` protocols, see Figure 6.2. The goal is to compute the User's key $K_U = \{w_1, w_2, \langle w_{3,i} \rangle_{i \in \mathcal{A}}\}$.

`Issuer_Att1` is run between the User and the Issuer. The User generates a cryptographic commitment $\bar{H} = h_1^{w_1} h_2^{w_2} \bmod p$ in $\mathbb{H}$, where the User's keys w_1, w_2 are committed values. Then, the User signs the commitment with his private key $sk_{\mathcal{U}}$ and sends it and the signature with the proof of construction PK_{U1} to the Issuer. The Issuer checks the signature and the proof and signs User's commitments by his private key $K_{\mathcal{I}}$. Commitments are stored by the Issuer for identification and revocation purposes. Any secure signature scheme, e.g. RSA, DSA, can be used.

`Issuer_Att2` is run between the User and the Manager. The User computes another commitment $\bar{A} = g_1^{w_1} g_2^{w_2} \bmod n$ in $\mathbb{OU}_n$ and sends $\bar{A}$, $\bar{H}$, the signature of $\bar{H}$ (generated by the Issuer) and the proof of discrete logarithm equivalence PK_{U2} to the Manager. Now, the Manager is able to compute the User's partial keys $\langle w_{3,i} \rangle_{i \in \mathcal{A}}$ for all attributes $\langle eca_i \rangle_{i \in \mathcal{A}}$ using the Equation 2.25 such that the following equalities hold:

$$\begin{aligned} eca_i &= w_1 \bullet G_1 + w_2 \bullet G_2 + w_{3,i} \bullet G_3 \\ &= w_1 \bullet G_1 + w_2 \cdot s_2 \bullet G_1 + w_{3,i} \cdot s_3 \bullet G_1 \\ &= (w_1 + w_2 \cdot s_2 + w_{3,i} \cdot s_3) \bullet G_1 \\ &= s_{1,i} \bullet G_1 = eca_i \end{aligned} \tag{6.1}$$

The values $\bar{A}, \bar{H}$ and $\langle w_{3,i} \rangle_{i \in \mathcal{A}}$ are stored in the Manager's database and sent to the User. The User securely stores his key $K_U = \{w_1, w_2, \langle w_{3,i} \rangle_{i \in \mathcal{A}}\}$, e.g. on a smart card.

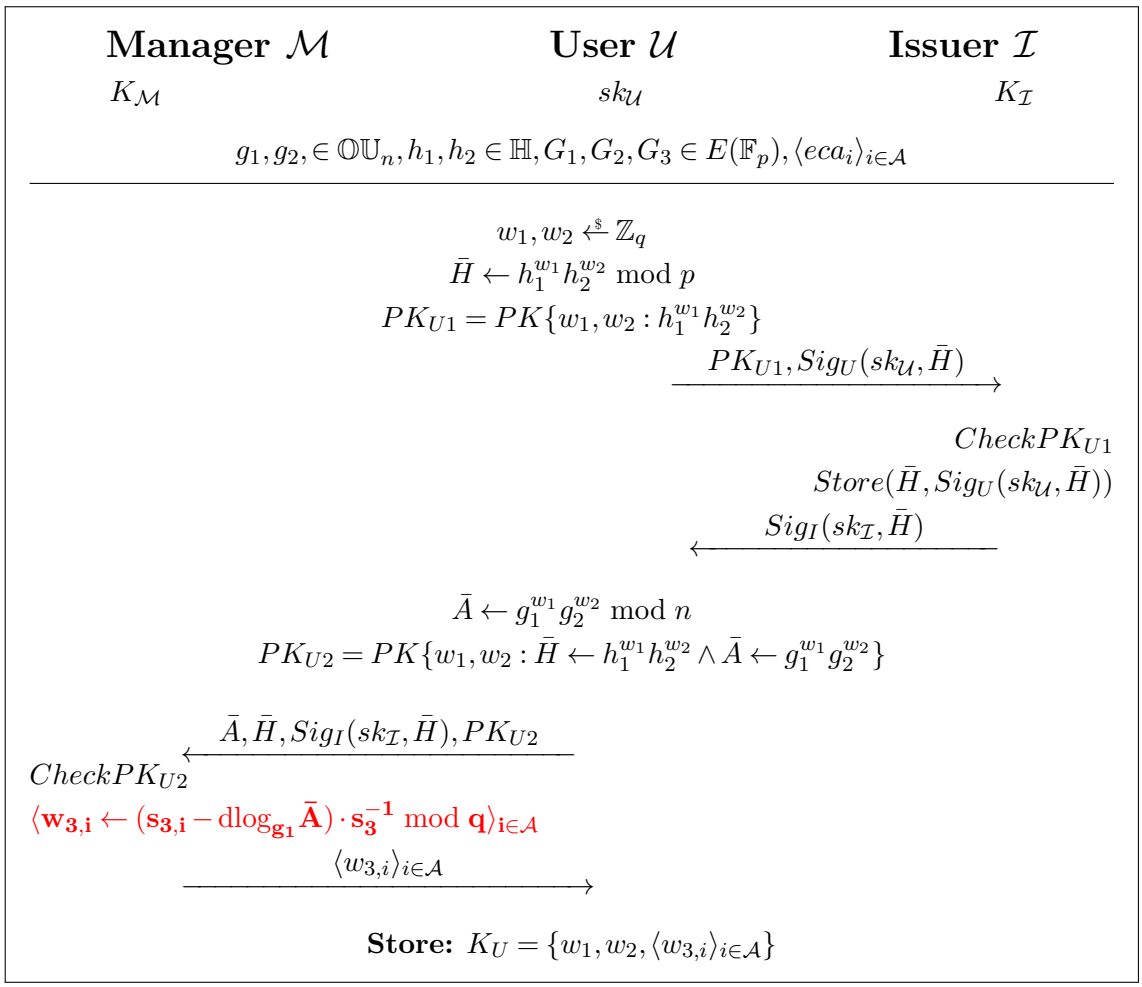

Fig. 6.2: `Issue_Att` protocol of the ecHM12 scheme.

`Prove_Att`

$(0/1) \leftarrow$ `Prove_Att`(par, K_U): this protocol is run fully over $E(\mathbb{F}_p)$. The protocol is depicted in Figure 6.3. The User proves the ownership of attributes $\langle eca_i \rangle_{i \in \mathcal{D}}$ to the Verifier using PK protocols. The unlinkability is provided by using the random number K_S, which is re-generated in every session. Moreover, the protocol provides revocation features by committing the value K_S in the commitment C_2 and the committed values $\langle w_{3,i} \rangle_{i \in \mathcal{D}}$ (revocable key parts of the User's key) in the commitments $\langle C_{1,i} \rangle_{i \in \mathcal{D}}$. The commitments $\langle C_{1,i} \rangle_{i \in \mathcal{D}}$ and C_2 permit to check if the User is in the blacklist or not, and to remove him from the system by involving the Manager in the revocation process. The verification time depends on the number of disclosed attributes by the User and on the number of all revoked Users.

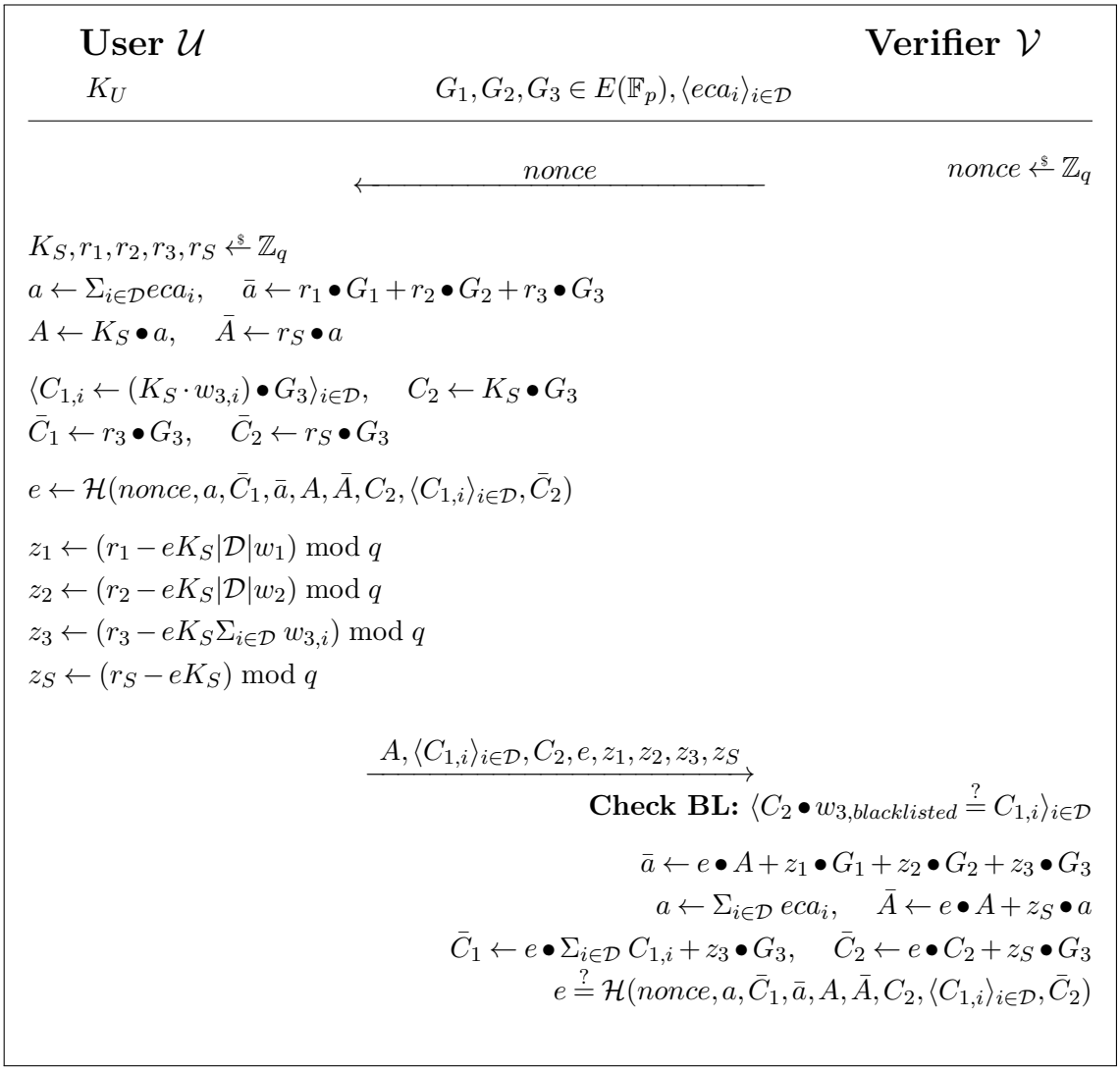

Fig. 6.3: `Prove_Att` protocol of the ecHM12 scheme.

Revoke

$(BL) \leftarrow$ `Revoke`$(par, proof, K_{\mathcal{M}})$: the original HM12 scheme uses the OU trapdoor to solve the discrete logarithm problem. In ecHM12 scheme, this trapdoor cannot be used. However, revocation of a dishonest user is still possible. The protocol input parameters are system parameters par and $proof$ generated by the User within the `Prove_Att` protocol. The revocation part of the $proof$ consists of commitments $\langle C_{1,i} \rangle_{i \in \mathcal{D}}$ and C_2. The Manager computes Equation 6.2 for all user keys $w_{3,DATABASE}$ in Manager's database until a match is found.

$$\langle w_{3,DATABASE} \bullet C_2 \stackrel{?}{=} C_{1,i} \rangle_{i \in \mathcal{R}} \tag{6.2}$$

If a match is found, the commitment that belongs to this particular User is revoked by publishing $\langle w_{3,i} \rangle_{i \in \mathcal{R}}$ (where $\mathcal{R}$ donates a subset of revoked attributes) on a Black List (BL). The revocation complexity is linear in the number of Users instead of constant as in the HM12 scheme. Yet revocation remains practical, see Section 6.5 for implementation details. On the other hand, the protocol `Prove_Att` is faster than in the HM12 scheme.

6.4 Security Analysis

The `ProveAtt` protocol is a standard proof of knowledge protocol that can be denoted as $PK\{(K_S w_1, K_S w_2, K_S \Sigma_{i \in \mathcal{D}}\, w_{3,i}, K_S) : A = K_S \bullet \Sigma_{i \in \mathcal{D}}\, eca_i \wedge A = K_S w_1 \bullet G_1 + K_S w_2 \bullet G_2 + K_S \Sigma_{i \in \mathcal{D}}\, w_{3,i} \bullet G_3\}$.

Completeness: (i.e., honest Users are always accepted by the protocol) is given by the design of the protocol and can be proven by expanding the Verifier's equations.

Soundness: (i.e., dishonest Users are always rejected by the protocol) is proven by employing the standard PK knowledge extractor that can extract $(K_S w_1, K_S w_2, K_S \Sigma_{i \in \mathcal{D}}\, w_{3,i}, K_S)$ and thus obtain valid user keys $(w_1, w_2, \langle w_{3,i} \rangle_{i \in \mathcal{D}})$. Thus, the `Prove_Att` protocol never accepts a User that does not know correct keys.

Zero-Knowledge: (i.e., the protocol does not release any information about the user's keys) is proven by creating the zero-knowledge simulator that can simulate the `ProveAtt` protocol. The simulator is constructed in the standard way, that is by choosing the answers z' in random and reconstructing the remaining values using the Verifier's equations:

$$(A', C'_1, C'_2) \xleftarrow{\$} E(\mathbb{F}_p), \quad (z'_1, z'_2, z'_3, z'_S, e') \xleftarrow{\$} \mathbb{Z}_q \tag{6.3}$$

$$\begin{aligned} \bar{a'} &= e \bullet A' + z'_1 \bullet G_1 + z'_2 \bullet G_2 + z'_3 \bullet G_3 \\ \bar{A'} &= e \bullet A' + z'_S \bullet a \\ \bar{C'}_1 &= e \bullet \Sigma_{i \in \mathcal{D}} C'_{1,i} + z'_3 \bullet G_3 \\ \bar{C'}_2 &= e \bullet C'_2 + z'_S \bullet G_3 \end{aligned} \tag{6.4}$$

The simulator's output $(A', \langle C'_{1,i} \rangle_{i \in \mathcal{D}}, C'_2, e', z'_1, z'_2, z'_3, z'_S)$ is indistinguishable from the real transcript of the `Prove_Att` protocol $(A, \langle C_{1,i} \rangle_{i \in \mathcal{D}}, C_2, e, z_1, z_2, z_3, z_S)$.

6.5 Implementation and Performance Analysis

ABC schemes are usually implemented using Java Card and MultOS smart card platforms. The Java Card platform lacks modular operations support as well as elliptic curve primitives support. Basic operations over elliptic curves are available on JC-3.0.5, but there is still no smart card with this operation system *available*. Therefore, we cannot consider Java Card in our tests. On the other hand, MultOS cards support elliptic curve points addition and scalar multiplication over elliptic curves. We use MultOS ML4 smart card to compare the HM12 standard scheme with the proposed ecHM12 scheme. We measured both schemes with comparable security level defined by NIST [2], i.e., 1392-bit version of HM12 and 160-bit version of ecHM12, and we also provided a comparison with 224-bit version of ecHM12 (higher level of security with respect to the previous values). On the smart card side, the comparison of the `Prove_Att` protocol is shown in Table 6.1. The ecHM12 scheme is faster than the HM12 scheme in the verification phase, even if a much higher security level of the ecHM12 scheme is used. Note that the efficiency of the verification phase is crucial for the scheme's speed, thus user friendliness. The elliptic curve scalar multiplication (ecMul) over $E(\mathbb{F}_{160})$ takes only 43 ms and 52 ms over $E(\mathbb{F}_{224})$ instead of 94 ms, that is the time required by modular exponentiation with 1392-bit base length and 560-bit exponent length in $\mathbb{Z}_n^*$. Data transmission is also improved: we need to transfer only 220 B in case of $E(\mathbb{F}_{160})$ or 308 B in case of $E(\mathbb{F}_{224})$ instead of 1558 B in the original scheme (1392-bit version) in the `Prove_Att` protocol. On the Verifier side, the time needed for checking blacklist is also more efficient in the ecHM12 scheme than in the HM12 scheme because of the involved operations: ecHM12 uses scalar multiplication and HM12 uses the slower modular exponentiation.

Tab. 6.1: Comparison of results in milliseconds for 1392-bit version of the HM12 scheme and equivalent 160-bit and 224-bit version of the proposed ecHM12 scheme.

		HM12 1392 bit		**ecHM12 160 bit**		**ecHM12 224 bit**	
Operation	Time [ms]	№	Time [ms]	№	Time [ms]	№	Time [ms]
modExp(160)	46	3	138	0	-	0	-
modExp(400)	72	2	150	0	-	0	-
modExp(560)	94	1	94	0	-	0	-
modExp(720)	112	2	224	0	-	0	-
modExp(880)	131	2	262	0	-	0	-
modMul	100	9	900	6	600	6	600
Sub	50	3	150	3	150	3	150
RNG	49	5	245	5	245	5	245
ecMul	52/48	0	-	10	480	10	520
ecAdd	25/23	0	-	2	46	2	50
Total	-	-	**2163**	-	**1521**	-	**1565**

Note: The key sizes are represented as bitlength and denote the minimal sizes for the given security strengths.

For the ecHM12 scheme, the revocation mechanism complexity is linear instead of constant as in the HM12 scheme. However, we expect Manager to be computationally strong and, consequently, the slow-down does not really affect the protocol complexity. We use oldish mid-range server, namely the 2009 IBM x3550 M2 with two Intel Xeon 2.27 GHz processors with 8 cores each and 32 GB RAM, to represent the Manager. The elliptic curve scalar multiplication over $E(\mathbb{F}_{224})$ took negligible 0.0189 ms, i.e. with 100,000 users in the system, the revocation time will be ca. 1.9 s at maximum.

In addition to the theoretical estimates presented above, we also provide full protocol implementation (proof of concept implementation). The protocol was implemented on MultOS ML4 smart card in 192-bit version (i.e. NIST curve P-192 was used). The implementation supports up to 5 attributes issued. The performance test results for increasing number of disclosed attributes are depicted in Figure 6.4. The time for one attribute possession proving takes around 1 s (including communication overhead) and ca. 2 s to prove possession of all 5 attributes.

Moreover, we provide comparison of our scheme implementation with implementation of the origin scheme HM12 (1024-bit version), see Chapter 3.2. The results show a significant time reduction (by 20% within on-card computation time and

almost by 40% in total, i.e. including communication between the card and the reader), since we use more efficient elliptic curve operations and transmit a smaller amount of data. Important to note, that our implementation holds significantly higher security level (1776-bit group equivalent according to [132] instead of 1024-bit group of the Idemix implementation). With increasing security strength of the protocols we can expect much bigger difference in attribute proving time and bandwidth usage.

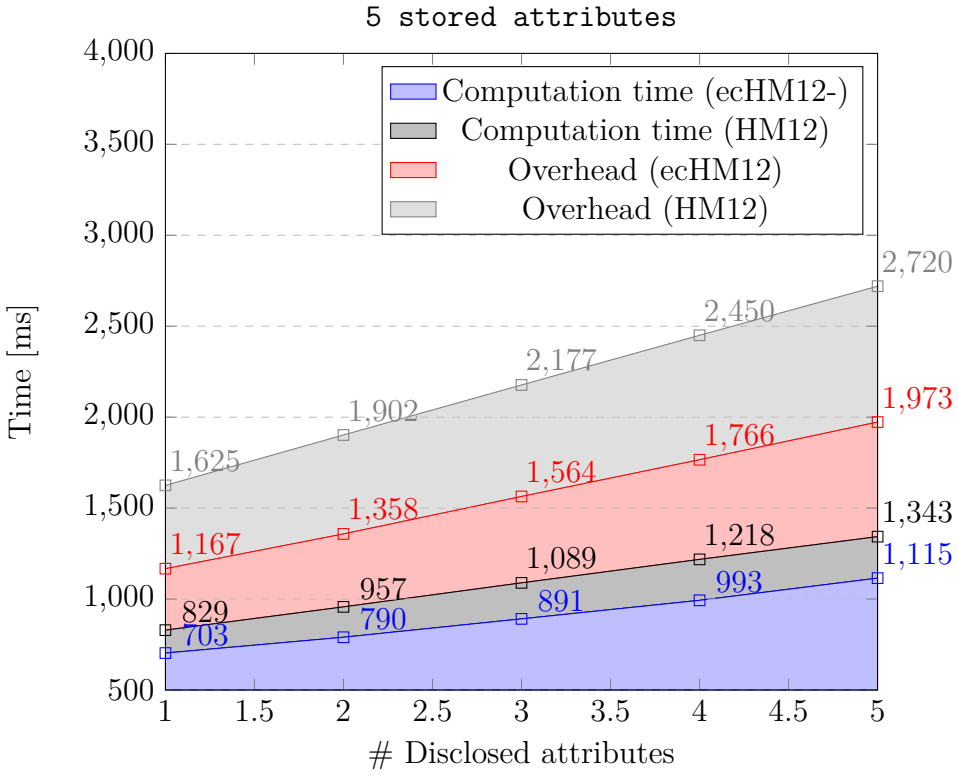

Fig. 6.4: Scheme ecHM12 attributes proving time for different scenarios.

6.6 Conclusion

We presented a new ABC scheme based on elliptic curves and the HM12 scheme. This variant meets all standard requirements on attribute-based credential schemes, i.e. anonymity, untraceability, unlinkability, selective disclosure of attributes, non-transferability, revocation and malicious user identification. By involving elliptic curves, the ecHM12 is faster in the `Prove_att` protocol, which makes the scheme more applicable in current access control systems. `Prove_att` protocol (on-card) is about 20% faster than in the HM12 scheme, and almost 40% faster considering also communication overhead. The efficiency advantage of our scheme grows with

higher security strength of the schemes. Our solution has also good impact on bandwidth, in fact, smaller amount of data is transferred. Data communication is 85% smaller compared to the HM12 protocol and considering a comparable security level (1392-bit in DL vs 160-bit EC security level).

The revocation process requires linear time in the number of Users instead of constant time of the HM12 scheme, but, considering that the current servers have high computing power, the slow-down does not really affect the protocol usability. Our next steps are the MultOS smart card optimization and the blacklist check optimization on the Verifier's side. Further, we would like to improve the complexity of the `Revoke` protocol.

7 Fast Keyed-Verification Anonymous Credentials

The content of this chapter have been published in conference paper [16].

7.1 Introduction

Anonymous credentials constitute a substantial part of privacy-enhancing cryptography. Using such credentials, users can anonymously prove the ownership of their personal attributes, such as age, nationality, sex or ticket validity. In the recent two decades, many proposals for anonymous credential schemes have been published. Starting with the fundamental works of Chaum [133], Brands [134], Camenish and Lysyanskaya [135], until recent schemes [68, 136, 137, 138, 139], researchers try to find a scheme that fulfills all requirements on privacy, is provably secure and is so efficient that it can be implemented on constrained devices. While there are schemes that fulfill all the requirements and can be implemented on PC and smartphone platforms, existing schemes deployed on smart cards are still not sufficiently fast for many applications, such as e-ticketing and eIDs. Yet, smart cards are the most appropriate platform for storing and proving personal attributes in everyday life, due to their size, security and reliability.

There are two major reasons why we lack practical implementations of anonymous credentials on smart cards. First, the complexity of asymmetric cryptographic algorithms used in anonymous credentials is quite high even for modern smart cards. Second, modern cryptographic schemes, including anonymous credentials, are mostly based on operations over an elliptic curve, while most available smart cards do not provide API for these operations. Particularly, the very popular operation of bilinear maps is still unsupported on this platform and simple operations, such as EC point scalar multiplication and addition, are significantly restricted.

In this chapter, we address both these concerns: First, we propose a novel keyed-verification anonymous credential scheme that is designed to allow for smart card implementations. Our scheme has the most efficient proving algorithm to date and requires only operations that are available on existing off-the-shelf smart cards. We present the implementation of our anonymous credential scheme that is 44 % - 72 % faster than the current state of the art implementations, while even providing a higher security level.

Second, we show that the practical implementation of EC-based schemes is not straightforward, due to the insufficient support of basic operations on most smart cards. We present the analysis of all major programmable smart card platforms

and show the availability and performance of EC operations. By showing which operations are available to developers on standard smart cards and how expensive those operations are, we hope to reduce the gap between protocol designers and developers. Using the analysis and benchmarks presented in this Chapter (and generally in this thesis), readers can decide whether their ECC-based scheme is implementable, choose the right platform and estimate its performance.

7.2 Related Work

Cryptographic anonymous credential schemes were first defined by the seminal works of Chaum [133], Brands [134] and Camenisch and Lysyanskaya [135]. The schemes were gradually improved by adding revocation protocols [140, 141], using more efficient algebraic structures [137, 142] and developing security models and formal proofs [143]. Idemix [113] and U-Prove [87] are the examples of the most evolved schemes aiming for a practical use. Recently, a new approach to obtain more efficient anonymous credentials schemes was proposed. Chase et al. [68] argue that in many scenarios where anonymous credentials could be deployed, the issuer of the credential will also serve as the verifier. This means that the verifier possesses the issuer key, which can be leveraged to obtain more efficient anonymous credential schemes tailored to setting. They formally define the so-called *Keyed-Verification Anonymous Credentials (KVAC)* and propose two instantiations. Barki et al. [144] propose a new KVAC scheme which is currently the most efficient: Proving possession of a credential with u hidden attributes costs $u + 12$ exponentiations.

Some of the new constructions were already implemented on the PC platform with promising results [145, 137]. Yet, the implementations on the smart card platform are available only for the former schemes that are based on traditional, rather inefficient modular structures [91, 1, 146, 121]. Furthermore, most implementations use only 1024-bit RSA groups that are considered insufficient by today's standards [132]. Implementations with higher security parameters [139, 136, 144, 147] either need distribution of computation to another device (usually a mobile phone) or use a non-standard proprietary API for EC operations and rely on pre-computations (which is impossible in crucial applications like e-ticketing and eID where the card is inactive and starts only for the attribute presentation). Regarding speed, the best-performing implementation of Idemix by the IRMA project [1] is able to compute the unlinkable attribute proof in at least 0.9 seconds, which is not convenient for time-critical applications where the proof should be presented in less than 500 ms. Currently, there is no cryptographic proposal and its implementation that would realize unlinkable anonymous credentials on the smart card platform with performance and security parameters necessary for a practical deployment.

Our Contribution

We propose a novel cryptographic scheme for anonymous attribute-based credentials that is designed primarily for smart cards. The scheme is based on our original algebraic MAC that makes its proving protocol very efficient. The computational complexity of our proving protocol is the lowest from related schemes (only $u+2$ scalar multiplications to present an attribute ownership proof) and we need only basic arithmetic operations that are already provided by existing smart cards' APIs. By analyzing the support of elliptic curve operations and their performance on current smart cards, we show that the implementation of even lightweight schemes is not easy on modern cards. We present the results of the full implementation of our proving protocol that is faster by at least 44 % than the state of the art implementation. By reaching the time of 366 ms including overhead, which is required for proving personal attributes on a 192-bit EC security level, we argue that the anonymous credentials are finally secure and practical even for time-critical and large-scale applications like eIDs, e-ticketing and mass transportation.

7.3 Cryptographic Design

We construct our keyed-verification anonymous credential scheme using the algebraic MAC scheme as presented in Section 7.3.1. In contrast to traditional anonymous attribute-based credential schemes, the verifier needs to know the secret keys to be able to verify user's attributes in keyed-verification anonymous credential schemes. This feature is particularly convenient for scenarios where attribute issuers and attribute verifiers are the same entities. The mass transportation settings is an example of such a scenario because the transportation authority both issues and checks the tickets and passes. The KVAC scheme supports all the standard privacy-enhancing features of ABC schemes, such as anonymity, unlinkability, untraceability, and selective disclosure of attributes, and is compatible with major credential schemes [113, 87] and standard revocation schemes [135, 66]. Although we primarily aim at keyed-verification credentials, our scheme can be easily enhanced to work also in the settings where the issuers are different from verifiers. Necessary modifications of cryptographic algorithms are described later in Section 7.3.6.

7.3.1 Our Algebraic MAC

This section describes our novel algebraic MAC scheme $\mathsf{MAC}_{\mathsf{wBB}}$, which is based on the weak Boneh-Boyen signature. It works in a group $\mathbb{G} = \langle g \rangle$ of prime order q with $|q| = \kappa$, as generated by $\mathtt{Setup}(1^\kappa)$. It can MAC vectors of n messages

$\vec{m} = (m_1, \ldots, m_n)$, with $m_i \in \mathbb{Z}_q$. The scheme is composed of the following algorithms.

`KeyGen`(par): chooses $x_i \xleftarrow{\$} \mathbb{Z}_q^*$ for $i = (0, \ldots, n)$ and outputs secret key $sk = (x_0, \ldots, x_n)$ and issuer parameters $ipar \leftarrow (X_0, \ldots, X_n)$ with $X_i = g^{x_i}$.

`MAC`($sk, \vec{m}$): let $sk = (x_0, \ldots, x_n)$ and $\vec{m} = (m_1, \ldots, m_n)$. The algorithm computes $\sigma = g^{\frac{1}{x_0 + \sum_{i=1}^{n} m_i x_i}}$ and auxiliary information $\sigma_{x_i} \leftarrow \sigma^{x_i}$ for $i = (1, \ldots, n)$.[1] Output the authentication code $(\sigma, \sigma_{x_1}, \ldots, \sigma_{x_n})$.

`Verify`($sk, \vec{m}, \sigma$): let $sk = (x_0, \ldots, x_n)$ and $\vec{m} = (m_1, \ldots, m_n)$. Then, the algorithm outputs 1 iff $g = \sigma^{x_0 + \sum_{i=1}^{n} m_i x_i}$.

Theorem 1. *Our MAC scheme is complete and unforgeable, as defined in Definition 1 and Definition 2, under the Strong DDH inversion assumption [148] and the delayed one-more DH problem [149].*

The proof was provided by IBM Research – Zurich and it is presented in the published paper [16].

7.3.2 Requirements

We require the scheme be provable secure and privacy-friendly (i.e. to provide anonymity, unlinkability and untraceability properties). Furthermore, we require that the scheme allows the user selectively disclose a set of his attributes, and the implementation should be efficient, thus it can be implemented on constrained devices such as smart cards. The detail description of the scheme requirements is provided below:

Security Requirements

- *Completeness*: if a User holds required attributes, he is always able to convince a Verifier about the claim truth and pass the `Prove_Att` protocol.
- *Soundness*: a User cannot pass the `Prove_Att` protocol without attribute possession that are required by the Verifier.
- *Provable security*: the scheme security is based on a computat ional hardness assumptions, see Chapter 2 for more details. Breaking the scheme is equivalent to solving hard mathematical problem that is assumed to be computationally impossible.

[1] Note that the the auxiliary information σ_{x_i} can be ommitted as they are not required for verification. However, in our keyed verification credentials that we introduce in the next section, it will turn out that adding these values will make credential presentation more efficient. For other uses of this MAC scheme, they can be freely ommitted for a more efficient MAC.

- *Zero-Knowledge*: the `Prove_Att` protocol transcript must be simulatable without the knowledge of the attributes, thus provably release no sensitive information.

Privacy Requirements

- *Anonymity*: each User anonymously proves possession of attributes. Therefore, his identity and his behaviour remains hidden in the system.
- *Untraceability*: all credentials are randomized, i.e., the issuer is not able to track user's movements and behaviour.
- *Unlinkability*: all single user's sessions are mutually unlinkable. Therefore, the verifier or an eavesdropper are not able to link individual sessions together and profile a user.
- *Selective disclosure of attributes*: a user can choose the attributes which have to be disclosed, other attributes remain hidden.

Efficiency Requirements

- *Readiness for smart card*: the scheme must be fast (efficient) on constrained devices, in particular smart cards. No operations, that are unavailable on smart cards (such as bilinear pairings), can be used in User's algorithms.

7.3.3 General Architecture

The communication pattern is presented in Figure 7.1 and employs:

- **User**: gets issued attributes from an Issuer and anonymously proves their possession to the Verifier.
- **Issuer**: is responsible for issuing attributes to a user. A issuer signs the user attributes with it (issuer) secret key.
- **Verifier**: verifies a possession of required attributes by the user. The verifier requires to have a issuer secret key, however, this necessity do not create any security risk, since we assume that the issuer and the verifier are the same entity.

Each entity communicates in the system through the specific cryptographic algorithms. Namelly, our KVAC scheme consists of algorithms (`Setup`, `CredKeygen`, `Issue`, `Obtain`, `Show`, `ShowVerify`)[2] that are executed by users and an issuer who also serves as a verifier.

[2] Note that Chase et al. [68] define BlindIssue and BlindObtain, but as we do not show efficient algorithms for blind issuance, we omit them from the definition here. Instead, we define `Obtain`, which lets a user check that a credential is indeed valid using only the public issuer parameters.

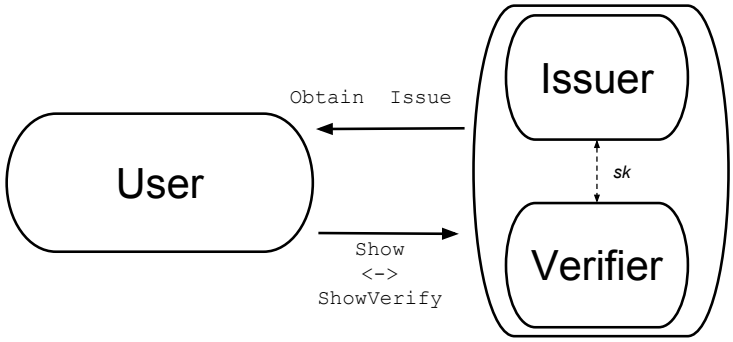

Fig. 7.1: Architecture of keyed-verification anonymous credentials.

$(par) \leftarrow \texttt{Setup}(1^\kappa)$: takes as input the security parameter κ and outputs the system parameters *par*. We will assume that *par* is given as implicit input to all other algorithms.

$(sk, ipar) \leftarrow \texttt{CredKeygen}(par)$: outputs a fresh issuer secret key *sk* and public issuer parameters *ipar*.

$(cred) \leftarrow \texttt{Issue}(sk, (m_1, \ldots, m_n))$: takes as input the issuer secret key and attribute values $(m_1, \ldots, m_n) \in \mathbb{Z}_q$ and outputs a credential *cred*.

$(0/1) \leftarrow \texttt{Obtain}(ipar, cred, (m_1, \ldots, m_n))$: lets a user verify a credential by giving as input the public issuer parameters, the credential and the attribute values.

$(0/1) \leftarrow [\texttt{Show}(ipar, cred, (m_1, \ldots, m_n), \phi) \leftrightarrow \texttt{ShowVerify}(sk, \phi)]$: is an interactive algorithm. The user runs `Show` on input the public issuer parameters, the credential, the attribute values and attribute predicate, and the verifier runs `ShowVerify` on input the issuer secret key and the attribute predicate, which will output 1 iff it accepts the credential presentation.

7.3.4 Cryptography Specification

In this section, we present our novel KVAC scheme that uses $\mathsf{MAC}_{\mathsf{wBB}}$ as introduced in Section 7.3.1. Our scheme is parametrized by n, the amount of attributes in a credential. We describe our scheme using *selective disclosure* as attribute predicates, i.e., a predicate ϕ can be seen as a set $\mathcal{D} \subseteq \{1, \ldots, n\}$ containing the indices of the disclosed attributes and the attribute values of the disclosed attributes $\langle m_i \rangle_{i \in \mathcal{D}}$. On a high level, we follow the approach from Chase et al [68] and build our KVAC scheme

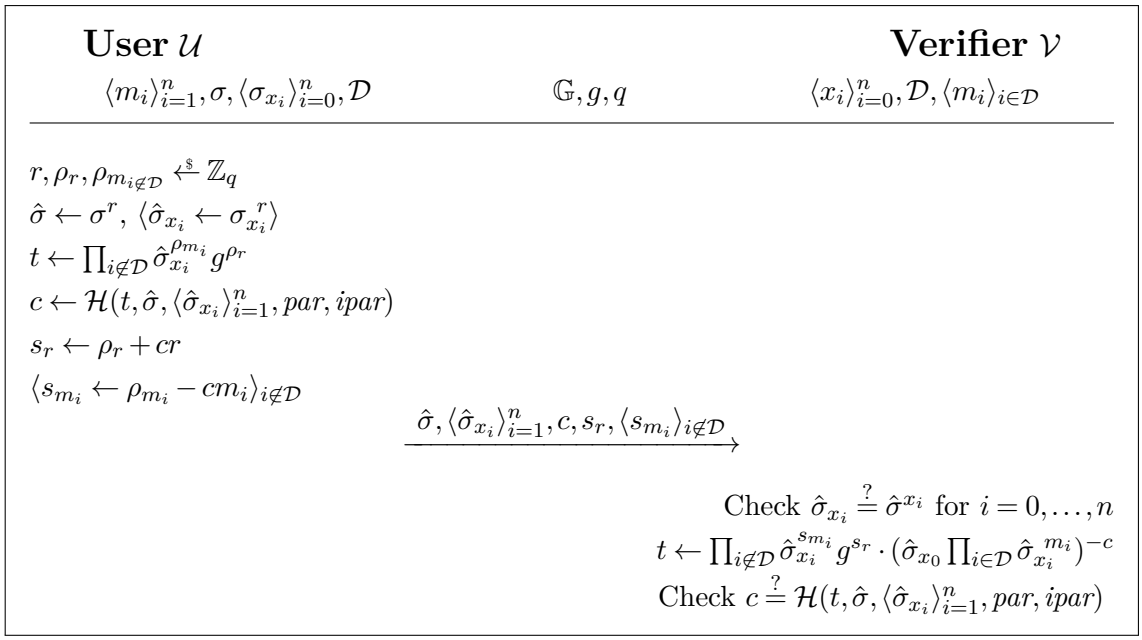

Fig. 7.2: Less efficient instantiation of `Show` and `ShowVerify` using a standard *SPK* proof, without optimizing for the fact that the issuer knows x_i.

from our algebraic MAC presented in Section 7.3.1 and zero knowledge proofs. One novel trick allows us to strongly improve the efficiency of our scheme. Instead of computing a standard noninteractive Schnorr-type proof of knowledge, we use the fact that the verifier knows the secret key. This allows us to omit elements that the verifier can compute by itself and saves the prover a lot of work.

We note that our `Issue` algorithm does not support the efficient issuance of committed attributes. This feature is useful in applications where a user needs to transfer his attributes among credentials or needs to get issued attributes that are only private to him. However, we consider these scenarios rare in targeted applications such as e-ticketing, mass transportation and loyalty cards. In those cases, the personal attributes (i.e., ticket type, pass validity period, registration number) are known to issuer or might even be chosen by the issuer. However, if the issuance of undisclosed attributes is necessary, it can be done by employing Paillier encryption [150], as is shown in [151].

`Setup`

$(par) \leftarrow$ `Setup`(1^κ): protocol outputs $par = (\mathbb{G}, g, q) \leftarrow$ `GroupSetup`(1^κ).

$(sk, ipar) \leftarrow$ `CredKeygen`(par): this protocol runs $(sk, ipar) \leftarrow \mathsf{MAC}_{\mathsf{wBB}}$.`KeyGen`$(par)$ and outputs sk and $ipar$.

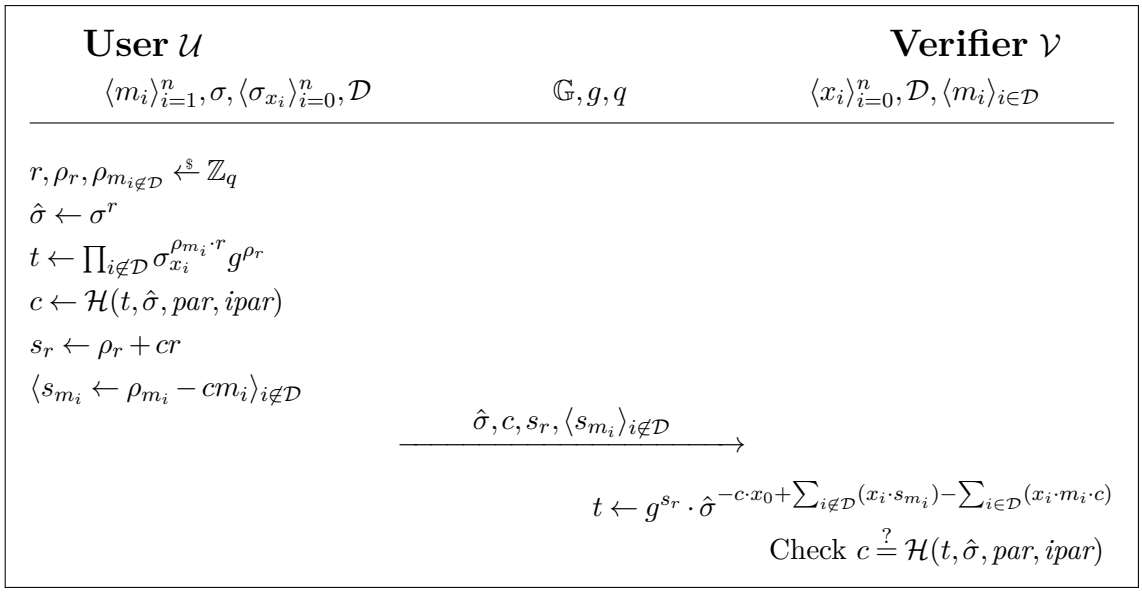

Fig. 7.3: Definition of the `Show` and `ShowVerify` algorithms of our KVAC scheme.

`Issue_Att`

$(cred) \leftarrow \texttt{Issue}(sk, (m_1, \ldots, m_n))$: runs $(\sigma, \langle\sigma_{x_i}\rangle_{i=0}^n) \leftarrow \mathsf{MAC}_{\mathsf{wBB}}.\texttt{MAC}(sk, (m_1, \ldots, m_n))$. Next, provides a proof that allows a user to verify the validity of the credential: $\pi \leftarrow SPK\{(x_0, \ldots, x_n) : \bigwedge_{i=0}^n \sigma_{x_i} = \sigma^{x_i} \wedge X_i = g^{x_i}\}$. The algorithm outputs credential $cred \leftarrow (\sigma, \langle\sigma_{x_i}\rangle_{i=0}^n, \pi)$.

$(0/1) \leftarrow \texttt{Obtain}(ipar, cred, (m_1, \ldots, m_n))$: parses $ipar$ as $(X_0, \ldots, X_n)$ and parses $cred$ as $(\sigma, \langle\sigma_{x_i}\rangle_{i=0}^n, \pi)$. The algorithm checks that $\sigma_{x_0} \cdot \prod_{i=1}^n \sigma_{x_i}^{m_i} = g$ and verifies π with respect to $ipar$ and σ.

`Prove_Att`

$(proof) \leftarrow \texttt{Show}(ipar, cred, (m_1, \ldots, m_n), (\mathcal{D}, \langle m_i\rangle_{i\in\mathcal{D}}))$: in credential presentation, we want to let the user prove posession of a valid credential with the desired attributes. On a high level, we want to prove knowledge of a weak Boneh-Boyen signature, so we can apply the efficient proof due to Arfaoui et al. [136] and Camenisch et al. [66], by extending it to support a vector of messages: Take a random $r \xleftarrow{\$} \mathbb{Z}_q$ and let $\hat{\sigma} \leftarrow \sigma^r$ and $\hat{\sigma}_{x_i} \leftarrow \sigma_{x_i}^r$ for $i = 0, \ldots, n$, and prove

$$proof = SPK\{(\langle m_i\rangle_{i\notin\mathcal{D}}, r) : \hat{\sigma}_{x_0} \prod_{i\in\mathcal{D}} \hat{\sigma}_{x_i}^{m_i} = g^r \prod_{i\notin\mathcal{D}} \hat{\sigma}_{x_i}^{-m_i}\}.$$

The verifier simply checks that the $\hat{\sigma}_{x_i}$ values are correctly formed and verifies the proof. This approach is depicted in Fig 7.2.

While this approach is secure and conceptually simple, it is not very efficient. We now present how we can construct a similar proof in a much more efficient manner. The key observation is that the user does not have to compute anything that the verifier, who is in possession of the issuer secret key sk, can compute. This means we can omit the computation of the $\hat{\sigma}_{x_i}$ values and define `Show` as follows. Randomize the credential by taking a random $r \leftarrow \mathbb{Z}_q^*$ and setting $\hat{\sigma} \leftarrow \sigma^r$. Take $\rho_r, \rho_{m_{i \notin \mathcal{D}}} \xleftarrow{\$} \mathbb{Z}_q$ and compute

$$t = \prod_{i \notin \mathcal{D}} \sigma_{x_i}^{\rho_{m_i} \cdot r} g^{\rho_r}, c = \mathcal{H}(t, \hat{\sigma}, par, ipar), s_r = \rho_r + cr, \langle s_{m_i} = \rho_{m_i} - cm_i \rangle_{i \notin \mathcal{D}}.$$

Send $(\hat{\sigma}, c, s_r, \langle s_{m_i} \rangle_{i \notin \mathcal{D}})$ to the verifier.

$(0/1) \leftarrow$ `ShowVerify`$(sk, (\mathcal{D}, \langle m_i \rangle_{i \in \mathcal{D}}), proof)$: the verifier running `ShowVerify` will receive $proof = (\hat{\sigma}, c, s_r, \langle s_{m_i} \rangle_{i \notin \mathcal{D}})$ from the user. It computes

$$t \leftarrow g^{s_r} \cdot \hat{\sigma}^{-c \cdot x_0 + \sum_{i \notin \mathcal{D}} (x_i \cdot s_{m_i}) - \sum_{i \in \mathcal{D}} (x_i \cdot m_i \cdot c)}$$

and checks that $c = \mathcal{H}(t, \hat{\sigma}, par, ipar)$. Output 1 if valid and 0 otherwise. The `Show` and `ShowVerify` algorithms are depicted in Figure 7.3.

Theorem 2. *Our keyed-verification credential scheme is secure following the definition by Chase et al. [68] (ommitting the blind issuance), under the Strong DDH Inversion assumption [148] and the Static DH problem [152], in the random oracle model.*

The proof was provided by IBM Research – Zurich and it is presented in the published paper [16].

7.3.5 Efficiency

Our `Show` and `ShowVerify` algorithms were designed to be efficient enough to run on smart cards. We avoided computing bilinear pairings due to their computational cost and the lack of support on existing smart cards. The use of the second most expensive operation, the exponentiation (or scalar multiplication of EC points respectively), is reduced to a minimum. Our proving algorithm, the part of the protocol we envision being executed on a smart card, only requires $u+2$ exponentiations, where u is the number of undisclosed attributes.

Table 7.1 compares the efficiency of our `Show` protocol to existing KVAC schemes [68, 139], well-known anonymous credential schemes U-Prove [87] and Identity Mixer [113], and a recent scheme by Ringers et al. [137]. Idemix takes place in the RSA group, meaning that the exponentiations are much more expensive than exponentiations in a prime order group. U-Prove lacks the unlinkability property. Compared

to $\mathsf{MAC}_{\mathsf{BB}}$, our scheme requires only 2 exponentiations without hidden attributes, whereas $\mathsf{MAC}_{\mathsf{BB}}$ requires 12, showing that especially for a small number of undisclosed attributes, our scheme is significantly faster than $\mathsf{MAC}_{\mathsf{BB}}$.

Tab. 7.1: Comparison of presentation protocols of credential schemes.

	Exp. prime	**Exp. RSA**	**Unlinkability**	**MAC**	**Security**
U-Prove [87]	$u+1$	0	✗	✗	-
Idemix [113]	0	$u+3$	✓	✗	sRSA [20]
Ringers et al. [137]	$n+u+9$	0	✓	✗	whLRSW [153]
$\mathsf{MAC}_{\mathsf{DDH}}$ [68]	$6u+12$	0	✓	✓	DDH [154]
$\mathsf{MAC}_{\mathsf{GGM}}$ [68]	$5u+4$	0	✓	✓	GGM [155]
$\mathsf{MAC}_{\mathsf{BB}}$ [139]	$u+12$	0	✓	✓	q-sDH [65]
Our work	$u+2$	0	✓	✓	sDDHI, SDH [148, 152]

7.3.6 Modifying our Scheme for Public Verification

Certain applications require *public* verification rather than keyed-verification, i.e., anybody with public parameters can verify attribute proofs instead of only the issuer. Our scheme can also be transformed to this setting. In that case, our scheme needs pairing friendly curves and place the X_i values in $\mathbb{G}_2$ rather than $\mathbb{G}_1$. Note that the zero knowledge proof in `Issue` can now be omitted, as the validity of $\hat{\sigma}_{x_i}$ can be checked using the bilinear pairing. To create proofs, we can use the non-optimized `Show` and `ShowVerify` algorithms as shown in Figure 7.2, but changing the checks on σ_{x_i} to use the pairing operation with the public issuer parameters instead of using the issuer secret key.

7.4 Security Analysis

The security analysis of the proposed scheme was provided by IBM Research – Zurich and it is fully presented in the published paper [16].

7.5 Implementation and Performance Analysis

This section contains the performance analysis of our scheme and the discussion regarding implementation aspects. In particular, we focus on problems with the

implementation of basic arithmetic operations on an elliptic curve on programmable smart cards. We show that even though a scheme is extremely efficient and using only standard operations, it is difficult to find a smart card that can be used for its practical implementation due to the lack of basic EC operations support and their insufficient performance.

7.5.1 Smart Card Selection

There are many cryptographic schemes for anonymous attribute-based credentials available. We analysed the most efficient ones in the Section 3.2. Nevertheless, the smart card implementations are only very few [91, 1, 88] and not practically usable as they use only small insecure security parameters to be able to achieve reasonable speed. Particularly, only 1024-bit RSA or DSA groups are used. That is considered insecure for any practical deployment today. In theory, replacing standard modular groups with elliptic curves would help with reducing security parameters size and improving speed. In practice, there are two major problems with the implementation of EC-based schemes on most modern cards: the basic arithmetic operations, particularly the EC point addition and scalar multiplication, are either not available on most cards (see Table 7.3) or they are very slow (see Figure 7.4).

We evaluated all major programmable smart card platforms, namely JavaCard, MultOS and BasicCards specified in Table 7.2.

For the implementation of most EC-based schemes, including ours, developers need EC point addition (`ecAdd`) and scalar multiplication (`ecMul`). Therefore, we analysed the support of these operations on selected smart cards. The results are provided in Table 7.3.

Tab. 7.2: Tested smart cards.

	J3A081	**J3D081**	**Sm@rtCafe6**	**Sm@rtCafe5**	**ZC7.6**	**ML4**	**ML3**
MCU	P5CD 081	P5CD 081	P5CD 081	P5CDs 080	–	SC23 Z018	SLE78 CLXPM
OS	JavaCard 2.2.2	JavaCard 3.0.1	JavaCard 3.0.1	JavaCard 2.2.2	Basic ZC7	MultOS 4.3.1	MultOS 4.3.1
ROM	264KB	264KB	264KB	200KB	–	252KB	280KB
EEPROM	80KB	80KB	80KB	80KB	72KB	18KB	96KB
RAM	6KB	6KB	6KB	6KB	4.3KB	1.75KB	2KB

Tab. 7.3: ECC support on tested smart cards.

	J3A081	J3D081	Sm@rtCafe6	Sm@rtCafe5	ZC7.6	ML4	ML3
ECDSA [b]	320	320	256	256	512	✗	512
ECDH [b]	320	320	256	256	512	✗	512
`ecAdd`	✓!	✓!	✗	✗	✓	✓	✗
`ecMul`	✓!	✓!	✗	✗	✓	✓	✗
`ecInv`	✗	✗	✗	✗	✗	✓	✗

Note: ✓– algorithm is fully supported, ✓!– algorithm is supported only through non-public JCOP API, ✗– algorithm is not supported.

Unfortunately, only 2 cards support the `ecAdd` and `ecMul` operations natively: MultOS ML4 and BasicCard ZC7.6. Other cards either do not support EC operations at all (.NET cards), support only part of them (some MultOS and JavaCard cards), or provide EC schemes like ECDH or ECDSA without any access to underlying arithmetic operations (typically JavaCards). This insufficient access to API is a crucial aspect for the implementation and should be considered by the designers and implementers of EC-based schemes. Therefore, due to their speed and support of all required operations, we selected the MultOS cards for our implementation. For detailed speed analysis of the most complex (but also the most common) operation, the scalar multiplication of EC points, see Figure 7.4.

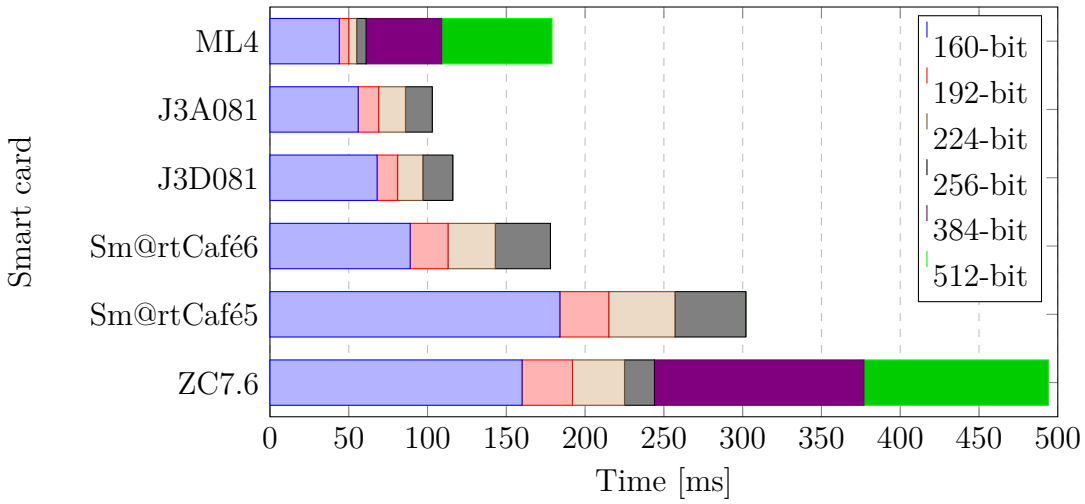

Fig. 7.4: Speed of EC point scalar multiplication operation on tested smart cards.

7.5.2 Implementation Results

The `Show` and `ShowVerify` algorithms of our scheme were implemented using a standard NIST P-192 curve [22]. We stress that this selection of parameters reflects contemporary recommendations regarding security levels, unlike other implementations of anonymous credentials that use mostly small modular groups. Only standard MultOS API and free public development environment (Eclipse IDE for C/C++ Developers, SmartDeck 3.0.1, MUtil 2.8) were used. For terminal application, Java BigInteger class and BouncyCastle API were used. We compare our results (blue and red) with the state of the art results of Vullers and Alpár (VA) [1] (black and white) for different numbers of attributes stored and disclosed in Figure 7.5. We note that our implementation uses significantly higher security parameters (1024-bit vs. 1776-bit DSA group equivalent according to [132]).

The algorithm time (blue) tells the time necessary to compute all algorithms on the card. The overhead time (red) adds time necessary to do all the supporting actions, mainly establishing the communication with a reader connected to PC and transferring APDUs. All results are arithmetic means of 10 measurements in milliseconds[3]. Compared to VA's implementation of Idemix, our implementation of all proving protocol algorithms on the card is at least 44% faster in all cases, see Figure 7.5 for details. In the case of only 2 attributes stored on the card, our scheme is by 72 % faster than VA's implementation. The card needs only 211 ms to compute the ownership proof for disclosed attributes. The total time of around 360 ms necessary for the whole proof generation on the card including communication with and computations on a terminal (standard PC, Core i7 2.4 GHz, 8 GB RAM) makes the implementation suitable also for time-critical applications like public transportation and ticketing. We also evaluated our scheme using an embedded device (Raspberry Pi 3) instead of the PC as a terminal. Even in that case the total time including overhead was below 450 ms. Based on our benchmarks, we expect that increasing security parameters to the 256-bit EC level would cost acceptable 15 % - 20 % in performance.

Our implementation is artificially limited to 10 attributes per a user, but the smart card's available memory resources (approx. 1.75 KB RAM and 7.5 KB usable EEPROM) would allow storing upto 50 attributes on a single card.

[3]Unlike microcontrollers and CPUs, smart card SDKs do not provide public tools for the measurement of clock cycles. Furthermore, the conversion between the number of cycles per an operation and it's execution time is difficult due to cards' variable clock speed. Therefore, the performance is usually measured in milliseconds [156, 91, 1, 146].

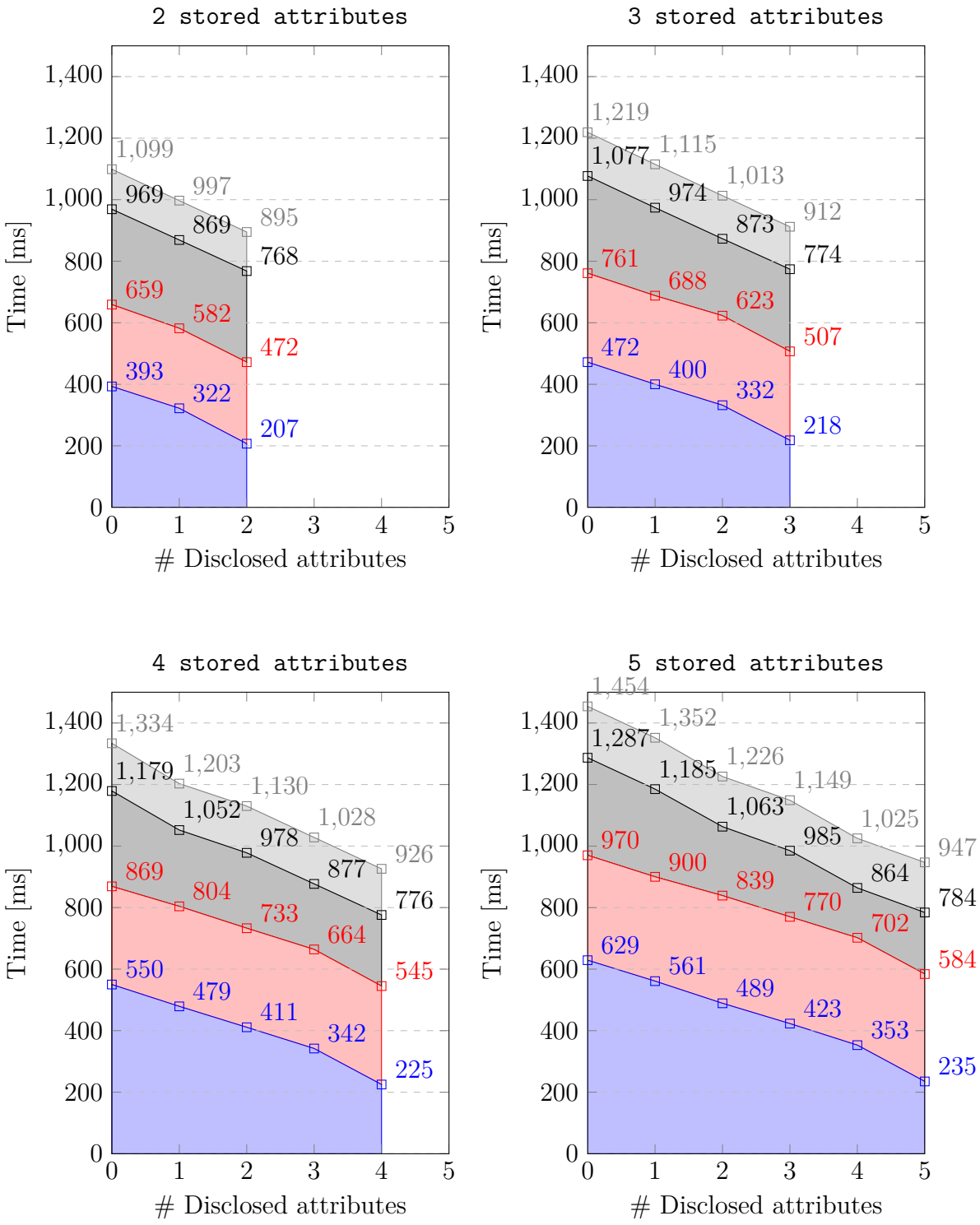

Fig. 7.5: Speed of our proving protocol compared to Vullers and Alpár (VA) implementation [1]. Blue - our algorithm time, red - our overhead, black - VA algorithm time and grey - VA overhead.

7.6 Conclusion

Practical anonymous credential schemes are only very few, with implementations on smart cards either too slow or providing insufficient security levels. Our approach to address this problem was twofold: 1) to propose a novel cryptographic scheme that is more efficient than all comparable schemes and formally prove its security; and 2) to develop a software implementation that is significantly faster than existing implementations, although they use lower security parameters. By achieving these results, we hope that we get privacy-enhanced authentication closer to practical applications.

Our future steps, besides further optimization, are the integration with a suitable revocation scheme (e.g., [66]) and implementation and benchmarks on higher security levels, hopefully on a wider range of smart cards, if they become available on the market.

8 Conclusion

The main goal of this doctoral thesis was to find novel privacy-preserving cryptographic solutions for current ICT application scenarios, especially for access control and data collection systems. The main emphasis was put on the support of new privacy-preserving features, such as anonymity, untraceability and unlinkability. Furthermore, the revocation and identification must remain possible and the developed schemes must be practical in wide applications, i.e. the implementation mus be efficient even on constrained devices, such as smart cards. Following these requirements, the thesis presents four novel lightweight privacy-preserving cryptographic proposals, that are provable secure and practical in many current ICT application scenarios.

The first proposed scheme, presented in Chapter 4, is provably secure and provides the full set of privacy-enhancing features, that is anonymity, untraceability and unlinkability of users. Furthermore, our scheme supports distributed multi-device authentication with multiple RFID user devices. This feature is particularly important in applications for controlling an access to dangerous areas where the presence of protective equipment is checked during each access control session. Besides the full cryptographic specification, we also show the results of our implementation on devices commonly used in access control applications, i.e. smart cards and embedded verification terminals. By avoiding costly operations on user devices, such as bilinear pairings, we were able to achieve times comparable with existing systems (around 500 ms), while providing significantly higher security, privacy protection and features for RFID multi-device authentication.

In Chapter 5, we provide the full cryptographic specification of our novel scheme for secure privacy-friendly data collection that is designed for computationally restricted user devices and supports all the security, privacy-protection and inspection features. Using the scheme, data can be anonymously collected from almost all types of devices, including simple sensors and smart meters. On the other side, malicious users can be efficiently identified and revoked. Furthermore, we provide the practical results of our implementation of the scheme on embedded devices, smart phones, smart cards, smart watches, computers and servers so that the efficiency can be thoroughly evaluated on various platforms.

Chapter 6 presents our novel anonymous attribute-based credential scheme. We modify the original scheme of Hajny and Malina [88] in a way that the scheme becomes more efficient due the use of elliptic curve construction. The scheme provides anonymity, untraceability, unlinkability, selective disclosure of attributes, non-transferability, revocation and malicious user identification as the original scheme. However, by involving elliptic curves, we achieved faster verification phase (by 30%)

and smaller communication cost between the user and the verifier (by 85%) compared to the original scheme, with equivalent or greater security level.

The last proposed scheme is presented in Chapter 7. The chapter introduces our novel keyed-verification credential system designed for lightweight devices (primarily smart cards) and provides security and efficiency proofs. By using a novel algebraic MAC based on Boneh-Boyen signatures, we achieve the most efficient proving protocol compared to existing schemes. In order to demonstrate the practicality of our scheme, we present an implementation on a standard, off-the-shelf, MultOS smart card. While using significantly higher security parameters than most existing implementations, we achieve performance that is more than 44 % better than the current state of the art implementations.

9 Selected Publications

In this chapter, we enumerate the publications that back the contents of this thesis:

1. Petr Dzurenda, Sara Ricci, Jan Hajny, and Lukas Malina. Performance analysis and comparison of different elliptic curves on smart cards. In International Conference on Privacy, Security and Trust (PST), pages 1–10, 2017. Calgary, Canada, ISBN: 978-1-5386-2487-6.
2. Radek Fujdiak, Petr Dzurenda, Petr Mlynek, Jiri Misurec, Milos Orgon, and Bezzateev Sergey. Anomalous behaviour of cryptographic elliptic curves over finite field. Elektronika ir Elektrotechnika, 23(5):82–88, 2017.
3. Jan Hajny, Petr Dzurenda, Sara Ricci, and Lukas Malina. Performance analysis of pairing-based elliptic curve cryptography on constrained devices. In In 2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Moscow, Russia, 2019.
4. Lukas Malina, Jan Hajny, Petr Dzurenda, and Vaclav Zeman. Privacy- preserving security solution for cloud services. Journal of applied research and technology, 13(1):20–31, 2015.
5. Lukas Malina, Petr Dzurenda, and Jan Hajny. Evaluation of anonymous digital signatures for privacy-enhancing mobile applications. International Journal of Security and Networks (online), 13(1):27–41, 2018.
6. Jan Hajny, Petr Dzurenda, and Lukas Malina. Attribute-based credentials with cryptographic collusion prevention. Security and Communication Networks, 8(18):3836–3846, 2015.
7. Jan Hajny, Petr Dzurenda, and Lukas Malina. Privacy-pac: Privacy-enhanced physical access control. In Proceedings of the ACM CCS, WPES '14, pages 93–96, New York, NY, USA, 2014. ACM.
8. Lukas Malina, Petr Dzurenda, Jan Hajny, and Zdenek Martinasek. Assessment of cryptography support and security on programmable smart cards. In 2018 41st International Conference on Telecommunications and Signal Processing (TSP), pages 1–5. IEEE, 2018.
9. Petr Dzurenda, Jan Hajny, Vaclav Zeman, and Kamil Vrba. Modern physical access control systems and privacy protection. In Telecommunications and Signal Processing (TSP), 2015 38th International Conference on, pages 1–5. IEEE, 2015.

10. Jan Hajny, Petr Dzurenda, and Lukas Malina. Multi-device authentication using wearables and iot. In SECRYPT 2016 Proceedings, ICETE 2016, pages 483–488, Portugal, 2016. SCITEPRESS - Science and Technology Publications, Lda. ISBN: 978-989-758-196-0

11. Jan Hajny, Petr Dzurenda, and Lukas Malina. Multidevice authentication with strong privacy protection. Wireless Communications and Mobile Computing, pages 1–12, 2018. vol. 2018, no. 3295148, ISBN: 1530-8669.

12. Jan Hajny, Petr Dzurenda, Lukas Malina, and Sara Ricci. Anonymous data collection scheme from short group signatures. In SECRYPT 2018 Proceedings, pages 1–10, 2018. ISBN: 978-989-758-319-3.

13. Petr Dzurenda, Jan Hajny, Lukas Malina, and Sara Ricci. Anonymous credentials with practical revocation using elliptic curves. In SECRYPT 2017 Proceedings, pages 534–539, 2017. ISBN: 978-989-758-259-2

14. Jan Camenish, Manu Drijvers, Petr Dzurenda, and Jan Hajny. Fast keyed-verification anonymous credentials on standard smart cards. In 34th International Conference on ICT Systems Security and Privacy Protection - IFIP SEC 2019, Lecture Notes in Computer Science (LNCS), Springer. 2019. Lisbon, Portugal, (Accepted).

Bibliography

[1] Pim Vullers and Gergely Alpar. Efficient selective disclosure on smart cards using idemix. In *Policies and Research in Identity Management*, volume 396 of *IFIP Advances in Information and Communication Technology*, pages 53–67. Springer Berlin Heidelberg, 2013.

[2] Elaine Barker. Recommendation for key management part 1: General (revision 4). *NIST Special Publication Part 1*, 800(57):1–147, 2016.

[3] Marr Bernard. How is big data used in practice? 10 use cases everyone must read, 2018.

[4] Juniper Research. Smart grids to save city dwellers $14bn in energy costs by 2022, 8th January 2018. Hampshire, UK.

[5] Andreas Poller, Ulrich Waldmann, Sven Vowé, and Sven Türpe. Electronic identity cards for user authentication-promise and practice. *IEEE Security & Privacy*, 10(1):46–54, 2012.

[6] Ministry of the interior of the Czech Republic. eObčanka: Začátek digitalizace v ČR, 2018.

[7] The White House. National strategy for trusted identities in cyberspace, enhancing online choice, efficiency, security, and privacy. Washington, April 14, 2011, Retrieved September 9, 2017. `https://www.nist.gov/sites/default/files/documents/2016/12/08/nsticstrategy.pdf`.

[8] Sonia Livingstone, Kjartan Ólafsson, and Elisabeth Staksrud. Social networking, age and privacy. The European Commission Safer Internet Programme, LSE Research Online, 2011. `http://eprints.lse.ac.uk/id/eprint/35849`.

[9] Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). *Official Journal of the European Union*, L119:1–88, May 2016.

[10] Ingo Naumann and Gilles Hogben. Enisa: Privacy features of eid cards. *Network Security Newsletter*, 2008:9–13, 2008.

[11] Communication from the commission to the european parliament, the council, the european economic and social committee and the committee of the regions

a european strategy on cooperative intelligent transport systems, a milestone towards cooperative, connected and automated mobility. *Official Journal of the European Union*, pages 1–12, 2016.

[12] Petr Dzurenda, Sara Ricci, Jan Hajny, and Lukas Malina. Performance analysis and comparison of different elliptic curves on smart cards. In *International Conference on Privacy, Security and Trust (PST)*, pages 1–10, 2017. Calgary, Canada, ISBN: 978-1-5386-2487-6.

[13] Jan Hajny, Petr Dzurenda, and Lukas Malina. Multidevice authentication with strong privacy protection. *Wireless Communications and Mobile Computing*, pages 1–12, 2018. vol. 2018, no. 3295148, ISBN: 1530-8669.

[14] Jan Hajny, Petr Dzurenda, Lukas Malina, and Sara Ricci. Anonymous data collection scheme from short group signatures. In *SECRYPT 2018 Proceedings*, pages 1–10, 2018. ISBN: 978-989-758-319-3.

[15] Petr Dzurenda, Jan Hajny, Lukas Malina, and Sara Ricci. Anonymous credentials with practical revocation using elliptic curves. In *SECRYPT 2017 Proceedings*, pages 534–539, 2017. ISBN: 978-989-758-259- 2.

[16] Jan Camenish, Manu Drijvers, Petr Dzurenda, and Jan Hajny. Fast keyed-verification anonymous credentials on standard smart cards. In *34th International Conference on ICT Systems Security and Privacy Protection - IFIP SEC 2019*, Lecture Notes in Computer Science (LNCS). Springer, 2019. Lisbon, Portugal, (Accepted).

[17] Jan Camenisch and Markus Stadler. *Efficient group signature schemes for large groups*, pages 410–424. Springer Berlin Heidelberg, 1997.

[18] Fré Vercauteren. Final report on main computational assumptions in cryptography, 2013. ECRYPT II, European Network of Excellence in Cryptology II, ICT-2007-216676.

[19] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[20] Ronald L. Rivest and Burt Kaliski. *RSA Problem*, pages 532–536. Springer US, Boston, MA, 2005.

[21] Patrik Bichsel, Carl Binding, Jan Camenisch, Thomas Groß, Tom Heydt-Benjamin, Dieter Sommer, and Greg Zaverucha. Specification of the identity mixer cryptographic library version 2.3.0*. Technical report, IBM, 2010.

[22] FIPS PUB 186-4. Federal Information Processing Standards Publication: Digital Signature Standard (DSS). 2013.

[23] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[24] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[25] Dan Boneh and Xavier Boyen. Short signatures without random oracles and the SDH assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, Apr 2008.

[26] Radek Fujdiak, Petr Dzurenda, Petr Mlynek, Jiri Misurec, Milos Orgon, and Bezzateev Sergey. Anomalous behaviour of cryptographic elliptic curves over finite field. *Elektronika ir Elektrotechnika*, 23(5):82–88, 2017.

[27] Lawrence Clinton Washington. *Elliptic curves: number theory and cryptography*. CRC press, 2008.

[28] Elaine B. Barker, Lily Chen, Allen Roginsky, and Miles Smid. NIST special publication 800-56a revision 2: Recommendation for pair-wise key establishment schemes using discrete logarithm cryptography. *National Institute of Standards and Technology*, 2013.

[29] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. Dhaes: An encryption scheme based on the diffie-hellman problem. *IACR Cryptology ePrint Archive*, 1999:7, 1999.

[30] Daniel R. L. Brown. Sec 1: Elliptic curve cryptography, version 2.0. *Standards for Efficient Cryptography*, pages 1–144, 2009.

[31] Neal Koblitz, Alfred Menezes, and Scott Vanstone. The state of elliptic curve cryptography. In *Towards a quarter-century of public key cryptography*, pages 103–123. Springer, 2000.

[32] Neal Koblitz. *Algebraic aspects of cryptography*, volume 3. Springer Science & Business Media, 2012.

[33] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren. *Handbook of elliptic and hyperelliptic curve cryptography*. CRC press, 2005.

[34] Nigel P. Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of cryptology*, 12(3):193–196, 1999.

[35] Daniel R. L. Brown. Sec 2: Recommended elliptic curve domain parameters, version 2.0. *Standards for Efficient Cryptography*, pages 1–33, 2010.

[36] Manfred Lochter and Johannes Merkle. Elliptic curve cryptography (ecc) brainpool standard curves and curve generation. Technical report, 2010.

[37] Benjamin Black, Joppe W Bos, Craig Costello, Patrick Longa, and Michael Naehrig. Elliptic curve cryptography (ecc) nothing up my sleeve (nums) curves and curve generation. Technical report, 2014.

[38] Jerome A. Solinas. Efficient arithmetic on koblitz curves. *Designs, Codes and Cryptography*, 19(2-3):195–249, 2000.

[39] Harold Edwards. A normal form for elliptic curves. *Bulletin of the American Mathematical Society*, 44(3):393–422, 2007.

[40] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. Twisted edwards curves. In *International Conference on Cryptology in Africa*, pages 389–405. Springer, 2008.

[41] Daniel J. Bernstein. Curve25519: new diffie-hellman speed records. In *International Workshop on Public Key Cryptography*, pages 207–228. Springer, 2006.

[42] Paulo SLM Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *International Workshop on Selected Areas in Cryptography*, pages 319–331. Springer, 2005.

[43] P.-Y. Liardet and Nigel P. Smart. Preventing spa/dpa in ecc systems using the jacobi form. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 391–401. Springer, 2001.

[44] Olivier Billet and Marc Joye. The jacobi model of an elliptic curve and side-channel analysis. In *International Symposium on Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes*, pages 34–42. Springer, 2003.

[45] John William Scott Cassels and Victor E. Flynn. *Prolegomena to a middlebrow arithmetic of curves of genus 2*, volume 230. Cambridge University Press, 1996.

[46] Nigel P. Smart. The hessian form of an elliptic curve. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 118–125. Springer, 2001.

[47] Marc Joye and Jean-Jacques Quisquater. Hessian elliptic curves and side-channel attacks. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 402–410. Springer, 2001.

[48] Daniel J Bernstein and Tanja Lange. Faster addition and doubling on elliptic curves. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 29–50. Springer, 2007.

[49] Huseyin Hisil, Kenneth Koon-Ho Wong, Gary Carter, and Ed Dawson. Faster group operations on elliptic curves. In *Proceedings of the Seventh Australasian Conference on Information Security-Volume 98*, pages 7–20. Australian Computer Society, Inc., 2009.

[50] Christophe Arene, Tanja Lange, Michael Naehrig, and Christophe Ritzenthaler. Faster computation of the tate pairing. *Journal of number theory*, 131(5):842–857, 2011.

[51] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 514–532. Springer, 2001.

[52] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Advances in Cryptology - CRYPTO'04*, 2004.

[53] Dan Boneh and Matt Franklin. Identity-based encryption from the weil pairing. In *Annual International Cryptology Conference*, pages 213–229. Springer, 2001.

[54] Kenneth G. Paterson. Id-based signatures from pairings on elliptic curves. *Electronics Letters*, 38(18):1025–1026, 2002.

[55] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[56] Ben Lynn. The pairing-based cryptography (pbc) library. `https://crypto.stanford.edu/pbc/`, 2018.

[57] M. U. Ltd. Multiprecision integer and rational arithmetic cryptographic library – the miracl crypto sdk. `https://github.com/miracl/MIRACL`, 2018.

[58] Akira Kanaoka. Tepla elliptic curve and pairing library. `https://github.com/TEPLA/tepla-library`, 2018. University of Tsukuba.

[59] Diego de Freitas Aranha and Conrado Porto Lopes Gouvêa. RELIC is an Efficient LIbrary for Cryptography. `https://github.com/relic-toolkit/relic`, 2018.

[60] Mitsunari Shigeo. Mcl library. `https://github.com/herumi/mcl`, 2018.

[61] Jan Hajny, Petr Dzurenda, Sara Ricci, and Lukas Malina. Performance analysis of pairing-based elliptic curve cryptography on constrained devices. In *In 2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Moscow, Russia, 2019.

[62] Ronald John Fitzgerald Cramer. *Modular Design of Secure Yet Practical Cryptographic Protocols*. PhD thesis, Universiteit van Amsterdam, 1997.

[63] Claus P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4:161–174, 1991.

[64] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

[65] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Advances in Cryptology - EUROCRYPT 2004*, pages 56–73, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[66] Jan Camenisch, Manu Drijvers, and Jan Hajny. Scalable revocation scheme for anonymous credentials based on n-times unlinkable proofs. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, WPES '16, pages 123–133, New York, NY, USA, 2016. ACM.

[67] Tatsuaki Okamoto and Shigenori Uchiyama. A new public-key cryptosystem as secure as factoring. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 308–318. Springer, 1998.

[68] Melissa Chase, Sarah Meiklejohn, and Greg Zaverucha. Algebraic MACs and keyed-verification anonymous credentials. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 1205–1216, New York, NY, USA, 2014. ACM.

[69] Yevgeniy Dodis, Eike Kiltz, Krzysztof Pietrzak, and Daniel Wichs. Message authentication, revisited. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of*

Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings, pages 355–374, 2012.

[70] David Chaum and Eugène Van Heyst. Group signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*, EUROCRYPT'91, pages 257–265, Berlin, Heidelberg, 1991. Springer-Verlag.

[71] Lukas Malina, Jan Hajny, Petr Dzurenda, and Vaclav Zeman. Privacy-preserving security solution for cloud services. *Journal of applied research and technology*, 13(1):20–31, 2015.

[72] Mark Manulis, Nils Fleischhacker, Felix Gunther, Franziskus Kiefer, and Bertram Poettering. Group signatures: Authentication with privacy. pages 1–267, 2012.

[73] Andreu Pere Isern-Deyà, Llorenç Huguet-Rotger, Magdalena M. Payeras-Capellà, and Macià Mut-Puigserver. On the practicability of using group signatures on mobile devices: implementation and performance analysis on the android platform. *International Journal of Information Security*, 14(4):335–345, 2015.

[74] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Annual International Cryptology Conference*, pages 255–270. Springer, 2000.

[75] Klaus Potzmader, Johannes Winter, Daniel Hein, Christian Hanser, Peter Teufl, and Liqun Chen. Group signatures on mobile devices: Practical experiences. In *International Conference on Trust and Trustworthy Computing*, pages 47–64. Springer, 2013.

[76] ISO/IEC 20008-2:2013. Information technology - security techniques - anonymous digital signatures - part 2: Mechanisms using a group public key. 2013. International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).

[77] Lukas Malina, Petr Dzurenda, and Jan Hajny. Evaluation of anonymous digital signatures for privacy-enhancing mobile applications. *International Journal of Security and Networks (online)*, 13(1):27–41, 2018.

[78] Cécile Delerablée and David Pointcheval. Dynamic fully anonymous short group signatures. In *Progress in Cryptology-VIETCRYPT 2006*, pages 193–210. Springer, 2006.

[79] Jung Yeon Hwang, Sokjoon Lee, Byung-Ho Chung, Hyun Sook Cho, and DaeHun Nyang. Short group signatures with controllable linkability. In *Lightweight Security & Privacy: Devices, Protocols and Applications (LightSec), 2011 Workshop on*, pages 44–52. IEEE, 2011.

[80] Jan Camenisch and Jens Groth. Group signatures: Better efficiency and new theoretical aspects. In *International Conference on Security in Communication Networks*, pages 120–133. Springer, 2004.

[81] Toshiyuki Isshiki, Kengo Mori, Kazue Sako, Isamu Teranishi, and Shoko Yonezawa. Using group signatures for identity management and its implementation. In *Proceedings of the second ACM workshop on Digital identity management*, pages 73–78. ACM, 2006.

[82] Jan Hajny, Lukas Malina, Zdenek Martinasek, and Vaclav Zeman. Privacy-preserving svanets: Privacy-preserving simple vehicular ad-hoc networks. In *Security and Cryptography (SECRYPT), 2013 International Conference on*, pages 1–8. IEEE, 2013.

[83] The Legion of the Bouncy Castle. Bouncy Castle Crypto APIs. 2013. `http://www.bouncycastle.org`.

[84] Angelo De Caro and Vincenzo Iovino. jpbc: Java pairing based cryptography. In *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011*, pages 850–855, Kerkyra, Corfu, Greece, June 28 - July 1, 2011. IEEE. `http://gas.dia.unisa.it/projects/jpbc/`.

[85] Gergely Alpár and Bart Jacobs. Credential design in attribute-based identity management. In *Leenes, R. (ed.), Bridging distances in technology and regulation, 3rd TILTing Perspectives Conference*. [Sl]: Wolf Legal Publishers, 2013.

[86] Pim Vullers. *Efficient implementations of attribute-based credentials on smart cards*. SI:[Sn], 2014.

[87] Christian Paquin and Greg Zaverucha. U-prove cryptographic specification v1.1 (revision 3). In *Microsoft*, pages 1–23, December 2013.

[88] Jan Hajny and Lukas Malina. Unlinkable attribute-based credentials with practical revocation on smart-cards. In Stefan Mangard, editor, *Smart Card Research and Advanced Applications - CARDIS 2012*, Lecture Notes in Computer Science, pages 62–76. Springer Berlin Heidelberg, 2013.

[89] David Pointcheval and Jacques Stern. Provably secure blind signature schemes. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 252–265. Springer, 1996.

[90] Claus Schnorr. Efficient identification and signatures for smart cards. In *Advances in cryptology—CRYPTO'89 proceedings*, pages 239–252. Springer, 1990.

[91] Wojciech Mostowski and Pim Vullers. Efficient U-Prove implementation for anonymous credentials on smart cards. In *International Conference on Security and Privacy in Communication Systems*, pages 243–260, Berlin, Heidelberg, 2011. Springer.

[92] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In *Proceedings of the 3rd international conference on Security in communication networks*, SCN'02, pages 268–289, Berlin, Heidelberg, 2003. Springer-Verlag.

[93] Jan Hajný. *Authentication Protocols and Privacy Protection.* Doctoral thesis, Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications, Brno, 2012.

[94] Gergely Alpár, Jaap H. Hoepman, and Wouter Lueks. An attack against fixed value discrete logarithm representations. *Cryptology ePrint Archive, Report 2013/120*, 2013. `http://eprint.iacr.org/`.

[95] Jan Hajny, Petr Dzurenda, and Lukas Malina. Attribute-based credentials with cryptographic collusion prevention. *Security and Communication Networks*, 8(18):3836–3846, 2015.

[96] Jan Hajny, Lukas Malina, and Ondrej Tethal. Privacy-friendly access control based on personal attributes. In *The 9th International Workshop on Security*, volume 8639 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2014.

[97] Jan Hajny, Petr Dzurenda, and Lukas Malina. Privacy-pac: Privacy-enhanced physical access control. In *Proceedings of the ACM CCS*, WPES '14, pages 93–96, New York, NY, USA, 2014. ACM.

[98] Lukas Malina, Petr Dzurenda, Jan Hajny, and Zdenek Martinasek. Assessment of cryptography support and security on programmable smart cards. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pages 1–5. IEEE, 2018.

[99] Daniel R. L. Brown. American national standards institute, "public key cryptography for the financial services industry: The elliptic curve digital signature algorithm (ecdsa)". *ANSI X9.62*, pages 1–55, 2005.

[100] WAP-261-WTLS-20010406-a. Specification information note. Technical report, Wireless Application Protocol Forum (WAP), 2001. Wireless Application Protocol WAP-261-WTLS-20010406-a Wireless Transport Layer Security.

[101] Konstantinos Markantonakis. Is the performance of smart card crytographic functions the real bottleneck? In *IFIP International Information Security Conference*, pages 77–91. Springer, 2001.

[102] Lukas Malina and Jan Hajny. Accelerated modular arithmetic for low-performance devices. In *Telecommunications and Signal Processing (TSP), 2011 34th International Conference on*, pages 131–135. IEEE, 2011.

[103] Petr Dzurenda, Jan Hajny, Vaclav Zeman, and Kamil Vrba. Modern physical access control systems and privacy protection. In *Telecommunications and Signal Processing (TSP), 2015 38th International Conference on*, pages 1–5. IEEE, 2015.

[104] Nicolas T. Courtois, Karsten Nohl, and Sean O'Neil. Algebraic attacks on the crypto-1 stream cipher in mifare classic and oyster cards. *IACR Cryptology ePrint Archive*, 2008.

[105] Nicolas T. Courtois. The dark side of security by obscurity and cloning mifare classic rail and building passes, anywhere, anytime. 2009.

[106] Flavio D. Garcia, Peter van Rossum, Roel Verdult, and Ronny Wichers Schreur. Wirelessly pickpocketing a mifare classic card. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 3–15. IEEE, 2009.

[107] FIPS PUB 197. Federal information processing standards publication: Advanced encryption standard (aes). *NIST*, 2001.

[108] David Oswald and Christof Paar. Breaking mifare desfire mf3icd40: Power analysis and templates in the real world. In *Cryptographic Hardware and Embedded Systems–CHES 2011*, pages 207–222. Springer, 2011.

[109] Konstantinos Markantonakis. Practical relay attack on contactless transactions by using nfc mobile phones. *Radio Frequency Identification System Security: RFIDsec*, 2012.

[110] Milosch Meriac. Heart of darkness-exploring the uncharted backwaters of hid iclasstm security. *Heart*, 2010.

[111] SRLabs. Legic prime rfid cards rely on obscurity and consequently did not withstand scrutiny. "`https://srlabs.de/analyzing-legic-prime-rfids/`", 2015. Security research labs.

[112] FIPS PUB 46-3. Federal information processing standards publication: Data encryption standard (des). *NIST*, 1999.

[113] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM conference on Computer and communications security*, CCS '02, pages 21–30, New York, NY, USA, 2002. ACM.

[114] Weitao Xu. Mobile applications based on smart wearable devices. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 505–506. ACM, 2015.

[115] Oriana Riva, Chuan Qin, Karin Strauss, and Dimitrios Lymberopoulos. Progressive authentication: Deciding when to authenticate on mobile phones. In *USENIX Security Symposium*, pages 301–316, 2012.

[116] Byung-Rae Cha, Sang-Hun Lee, Soo-Bong Park, Gun-Ki Lee4 Yoo-Kang Ji, et al. Design of micro-payment to strengthen security by 2 factor authentication with mobile & wearable devices. *Advanced Science and Technology Letters*, 109(7):28–32, 2015.

[117] Ismail Butun, Melike Erol-Kantarci, Burak Kantarci, and Houbing Song. Cloud-centric multi-level authentication as a service for secure public safety device networks. *IEEE Communications Magazine*, 54(4):47–53, April 2016.

[118] Lorena Gonzalez-Manzano, José María de Fuentes, and Agustín Orfila. Access control for the cloud based on multi-device authentication. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 856–863, Aug 2015.

[119] Jan Hajny, Petr Dzurenda, and Lukas Malina. Multi-device authentication using wearables and iot. In *SECRYPT 2016 Proceedings*, ICETE 2016, pages 483–488, Portugal, 2016. SCITEPRESS - Science and Technology Publications, Lda. ISBN: 978-989-758-196-0.

[120] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *Proceedings of the 11th ACM Conference on Computer and Communications Security*, CCS '04, pages 132–145, New York, NY, USA, 2004. ACM.

[121] Patrik Bichsel, Jan Camenisch, Thomas Groß, and Victor Shoup. Anonymous credentials on a standard java card. In *Proceedings of the 16th ACM conference on Computer and communications security*, CCS '09, pages 600–610, New York, NY, USA, 2009. ACM.

[122] Dan Boneh and Hovav Shacham. Group signatures with verifier-local revocation. In *Proceedings of the 11th ACM conference on Computer and communications security*, CCS '04, pages 168–177, New York, NY, USA, 2004. ACM.

[123] Anna Lisa Ferrara, Matthew Green, Susan Hohenberger, and Michael Østergaard Pedersen. Practical short signature batch verification. In *Topics in Cryptology - The Cryptographers' Track at the RSA Conference*, volume 5473, pages 309–324. Springer, April 2009.

[124] Kitae Kim, Ikkwon Yie, Seongan Lim, and Daehun Nyang. Batch verification and finding invalid signatures in a group signature scheme. *IJ Network Security*, 13(2):61–70, 2011.

[125] Xiaodong Lin, Xiaoting Sun, Pin han Ho, and Xuemin Shen. Gsis: A secure and privacy preserving protocol for vehicular communications. In *IEEE Transactions on Vehicular Technology*, volume 56, pages 3442–3456, 2007.

[126] Chenxi Zhang, Rongxing Lu, Xiaodong Lin, Pin-Han Ho, and Xuemin Shen. An efficient identity-based batch verification scheme for vehicular sensor networks. In *INFOCOM*, pages 246–250. IEEE, 2008.

[127] David Pointcheval and Olivier Sanders. *Short Randomizable Signatures*, pages 111–126. Springer International Publishing, Cham, 2016.

[128] Soren Finster and Ingmar Baumgart. Pseudonymous smart metering without a trusted third party. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, pages 1723–1728, July 2013.

[129] Cristina Rottondi, Giulia Mauri, and Giacomo Verticale. A protocol for metering data pseudonymization in smart grids. *Transactions on Emerging Telecommunications Technologies*, 26(5):876–892, 2015.

[130] Maxim Raya and Jean-Pierre Hubaux. Securing vehicular ad hoc networks. *J. Comput. Secur.*, 15:39–68, January 2007.

[131] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of group signatures: The case of dynamic groups. In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, pages 136–153, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[132] Nigel Smart. Ecrypt ii yearly report on algorithms and keysizes. Technical report, Katholieke Universiteit Leuven. ECRYPT II European Network of Excellence in Cryptology II, ICT-2007-216676.

[133] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, October 1985.

[134] Stefan A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, Cambridge, MA, USA, 2000.

[135] Jan Camenisch and Anna Lysyanskaya. *An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation*, pages 93–118. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[136] Ghada Arfaoui, Jean-François Lalande, Jacques Traoré, Nicolas Desmoulins, Pascal Berthomé, and Saïd Gharout. A practical set-membership proof for privacy-preserving NFC mobile ticketing. *Proceedings on Privacy Enhancing Technologies*, 2015(2):25–45, 2015.

[137] Sietse Ringers, Eric R. Verheul, and Jaap-Henk Hoepman. An efficient self-blindable attribute-based credential scheme. *IACR Cryptology ePrint Archive*, 2017:115, 2017.

[138] Gesine Hinterwälder, Felix Riek, and Christof Paar. Efficient e-cash with attributes on Multos smartcards. In *RFIDSec*, 2015.

[139] Amira Barki, Sollen Brunet, Nicolas Desmoulins, and Jacques Traoré. Improved algebraic MACs and practical keyed-verification anonymous credentials. In *Proceedings of the 2016 Selected Areas in Cryptography - SAC 2016*, 2016.

[140] Jan Camenisch and Anna Lysyanskaya. *Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials*, pages 61–76. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[141] Jan Camenisch, Markulf Kohlweiss, and Claudio Soriente. *Solving Revocation with Efficient Update of Anonymous Credentials*, pages 454–471. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[142] Jan Camenisch, Gregory Neven, and Markus Rückert. *Fully Anonymous Attribute Tokens from Lattices*, pages 57–75. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[143] Jan Camenisch, Stephan Krenn, Anja Lehmann, Gert Læssøe Mikkelsen, Gregory Neven, and Michael Østergaard Pedersen. Scientific comparison of ABC protocols. 2014.

[144] Amira Barki, Solenn Brunet, Nicolas Desmoulins, Sébastien Gambs, Saïd Gharout, and Jacques Traoré. *Private eCash in Practice (Short Paper)*, pages 99–109. Springer Berlin Heidelberg, Berlin, Heidelberg, 2017.

[145] Marios Isaakidis, Harry Halpin, and George Danezis. UnlimitID: Privacy-Preserving Federated Identity Management Using Algebraic MACs. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, WPES '16, pages 139–142, New York, NY, USA, 2016. ACM.

[146] Antonio de la Piedra, Jaap-Henk Hoepman, and Pim Vullers. *Towards a Full-Featured Implementation of Attribute Based Credentials on Smart Cards*, pages 270–289. Springer International Publishing, Cham, 2014.

[147] Amira Barki, Nicolas Desmoulins, Saïd Gharout, and Jacques Traoré. Anonymous attestations made practical. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec '17, pages 87–98, New York, NY, USA, 2017. ACM.

[148] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pages 201–210, 2006.

[149] Neal Koblitz and Alfred Menezes. Another look at non-standard discrete log and diffie-hellman problems. *J. Mathematical Cryptology*, 2(4):311–326, 2008.

[150] Pascal Paillier. *Public-Key Cryptosystems Based on Composite Degree Residuosity Classes*, pages 223–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[151] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. *Randomizable Proofs and Delegatable Anonymous Credentials*, pages 108–125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[152] Daniel R. L. Brown and Robert P. Gallant. The static diffie-hellman problem. *IACR Cryptology ePrint Archive*, 2004:306, 2004.

[153] Victor K. Wei and Tsz Hon Yuen. More short signatures without random oracles. Cryptology ePrint Archive, Report 2005/463, 2005. `https://eprint.iacr.org/2005/463`.

[154] Dan Boneh. The decision diffie-hellman problem. In *Algorithmic Number Theory*, pages 48–63, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[155] Victor Shoup. *Lower Bounds for Discrete Logarithms and Related Problems*, pages 256–266. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[156] Jan Hajny, Lukas Malina, Zdenek Martinasek, and Ondrej Tethal. *Performance Evaluation of Primitives for Privacy-Enhancing Cryptography on Current Smart-Cards and Smart-Phones*, pages 17–33. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

List of Symbols, Physical Constants and Abbreviations

AAM	Application Abstract Machine
ABC	Attribute-Based Credential
ACJT	Ateniese-Camenisch-Joye-Tsudik signature
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
APDU	Application Protocol Data Units
API	Application Programming Interface
BBS	Boneh-Boyen-Shacham signature
BL	Black List
BLE	Bluetooth Low Energy
BN	Barreto-Naehrig Curves
BSI	German Federal Office for Information Security
CDH	Computational Diffie-Hellman
C-ITS	European Strategy on Cooperative Intelligent Transport Systems
CPU	Central Processing Unit
CS	Camenisch and Stadler
DDH	Decision Diffie-Hellman
DES	Data Encryption Standard
DH	Diffi-Hellman
DL	Discrete Logarithm
DLP	Discrete Logarithm Problem
DPA	Differential Power Analysis
DSA	Digital Signature Algorithm
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
ECMQV	Elliptic Curve Menezes-Qu-Vanstone
EEPROM	Electrically Erasable Programmable Read-Only Memory
eIDs	electronic IDentity cards
ENISA	European Network and Information Security Agency
EU	European Union
FIPS	Federal Information Processing Standards

GDPR	General Data Protection Regulation
GMP	GNU Multi-Precision Arithmetic Library
GPS	Global Positioning System
HIV	Human Immunodeficiency Virus
HM12	Hajny-Malina
HVZK	Honest Verifier Zero-Knowledge
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
IFP	Integer Factorization Problem
IoT	Internet of Thinks
ISO	International Organization for Standardization
JC	Java Card
JCOP	Java Card OpenPlatform
JCVM	Java Card Virtual Machine
JPBC	Java Pairing-Based Cryptography
KVAC	Keyed-Verification Anonymous Credentials
MAC	Message Authentication Codes
MCU	Microcontroller Unit
MEL	MULTOS Executable Language
MIRACL	Multiprecision Integer and Rational Arithmetic Cryptographic Library
MUtil	MultOS Utility
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
NUMS	Nothing Up My Sleeve
NXP	Next eXPerience
OS	Operating System
OU	Okamoto-Uchiyama public-key encryption scheme
PBC	Pairing Based Cryptography
PI	Prover Information
PK	Proof of Knowledge
PKDLR	Proof of Knowledge Discrete Logarithm Representation
PPT	Probabilistic Polynomial Time
RAM	Random Access Memory
RELIC	Efficient LIbrary for Cryptography
RFID	Radio-Frequency IDentification
ROM	Read-Only Memory
RSA	Rivest, Shamir and Adleman
SCADA	Supervisory Control And Data Acquisition
SDDH	Strong Decision Diffie-Hellman

SDDHI	Strong Decision Diffie-Hellman Inversion
SDH	Static Diffie-Hellman
SDK	Software Development Kit
SECG	Standards for Efficient Cryptography Group
SHA	Secure Hash Algorithm
SIN	Social Insurance Number
SPA	Simple Power Analysis
SPK	Signature Proof of Knowledge
SRSA	Strong Rivest, Shamir and Adleman
TEPLA	University of Tsukuba Elliptic Curve and Pairing Library
TTP	Third Trusted Party
TV	Television
UF-CMVA	Unforgeability under a Chosen Message and Verification Attack
UID	User IDentifier
US	United States
VANET	Vehicular Ad hoc Networks
VLR	Verifier Local Revocation
wBB	weak Boneh-Boyen Signature
WTLS	Wireless Transport Layer Security
XOR	eXclusive OR
ZC	ZeitControl

List of Appendices

A Appendix: Idemix Security Parameters

Tab. A.1: System parameter sizes (in bits) used in Idemix scheme according to the security parameter $\kappa = 160$ that corresponds to **Security Strength = 80** defined by NIST [2].

Parameter	**Bitlength**	**Note**
l_n	1024	RSA Modulus Size, $n = pq$
l_p	512	$p = 2p_{SG} + 1, \quad p, p_{SG} \bigcap \mathbb{P}$
l_q	512	$q = 2q_{SG} + 1, \quad q, q_{SG} \bigcap \mathbb{P}$
l'_e	120	$l'_e < l_e - l_\phi - l_H - 3$
l_m, l_a	160	User Secret Size, Attributes Size
l_ϕ	80	Error Size (source name `l_statzk`)
l_H	160	Hash Size, $l_H < l_e, \quad l_H \geq \kappa$
l_v	1508	$l_n + l_\phi + l_H + \max\{l_m + l_r + 3, l_\phi + 2\} + 1$
l_e	405	$l_\phi + l_H + \max\{l_m + 4, l'_e + 2\} + 1$, $e \bigcap \mathbb{P}$
$l_{v'}$	1104	$l_n + l_\phi$
l_{n_1}, l_{n_2}	80	l_ϕ
$l_{\widetilde{v'}}$, $l_{\widetilde{r}}$	1344	$l_n + 2l_\phi + l_H$
$l_{v''}$	1508	l_v
l_{r_A}	1104	$l_n + l_\phi$
$l_{\widetilde{e}}$	360	$l'_e + l_\phi + l_H$
$l_{\widetilde{v}}$	1748	$l_v + l_\phi + l_H$
$l_{\widetilde{m}}$, $l_{\widetilde{a}}$	401	$l_m + l_\phi + l_H + 1$

B Appendix: HM12 Security Parameters

Tab. B.1: System parameter sizes (in bits) used in HM12 scheme according to the security parameter $\kappa = 160$ that corresponds to **Security Strength = 80** defined by NIST [2].

Parameter	**Bitlength**	**Note**
l_H	160	Hash Size
l_a	80	User Secret Size
l_ϕ	80	Error Size
l_p	1024	$p \bigcap \mathbb{P}$
l_q	160	$q \vert p-1,\ 2l_a$
l_n	1024	OU Modulus Size, $n = r^2 s$
l_r	360	$r = 2r'+1, \quad r, r' \bigcap \mathbb{P}, \quad l_r \geq 4.5 l_a$
l_s	304	$s = 2s'+1, \quad s, s' \bigcap \mathbb{P}$
l_{w_1}	160	$2l_a$
l_{w_2}	80	l_a
l_{w_3}	352	$l_{w_3} < l_r$
l_{S_1}	200	$2.5 l_a$
l_{S_2}	80	l_a
$l_{S_3^{-1}}$	80	l_a
l_{K_S}	160	$2l_a$
l_{r_1}	560	$l_\phi + l_H + l_{K_S} + l_{w_1}$
l_{r_2}	480	$l_\phi + l_H + l_{K_S} + l_{w_2}$
l_{r_3}	732	$l_\phi + l_H + l_{K_S} + l_{w_3}$
l_{r_S}	400	$l_\phi + l_H + l_{K_S}$

Ing. Petr **Dzurenda**

Researcher · Academic staff

Brno University of Technology,
Faculty of Electrical Engineering and Communication, Department of Telecommunications,
Technicka 12, 616 00, Brno,
Czech Republic

(+420) 720 591 809 | dzurenda@feec.vutbr.cz | https://www.vutbr.cz/lide/petr-dzurenda-106420

Professional experience

2014 - Present	**Scientific Worker** *at Centre of Sensor, Information and Communication Systems (SIX).*
2014 - Present	**Researcher** *at Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications.*
2014 - Present	**Academic staff** *at Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications.*

Education

2013 - Present	**Doctoral program (Ph.D.)** *at Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications.* *Thesis title: Cryptographic protection of digital identity*
2010 - 2013	**Master program (Ing.)** *at Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications.* *Thesis title: The security risks of authentication methods*
2007 - 2010	**Bachelor program (Bc.)** *at Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications.* *Thesis title: Data transmition security with general linear block codes*

Internship abroad

10/2016 - 01/2017	**Universitat Rovira i Virgili,** *Department of Computer Engineering and Mathematics, Tarragona, Spain. Research in: Elliptic Curve Cryptography, Performance of different elliptic curves on Smart Cards. Duration: four month*
09/2015 - 12/2015	**Universitat Rovira i Virgili,** *Department of Computer Engineering and Mathematics, Tarragona, Spain. Research in: Privacy Enhancing Technologies, Privacy in Low Emission Zones and Automatic Fare Collection systems. Duration: four month*

Publication Activities (Scopus, Web of Science)

IF journals	*5*
Technical journals	*7*
International conf.	*18*
Total publications	*30*
H-index (WoS)	*1*
H-index (Scopus)	*3*
H-index (Google Scholar)	*4*

Research projects

2018 - 2020	**Strategic Programs for Advanced Research and Technology in Europe - SPARTA (H2020-SU-ICT-03-2018):** *Horizon 2020: Research and Innovation action (RIA)*

2019 - 2021	**Legal and technical means of privacy protection in cyberspace (TL02000398):** *Technology Agency of the Czech Republic (TACR)*
2017 - 2022	**Creation of a double-degree doctoral study program Electronics and Information Technology and creation of a doctoral study program Information Security - OP3V (CZ.02.2.69/0.0/0.0/16_018/0002575):** *Ministry of Education, Youth and Sports (MSMT)*
2017 - 2020	**Automated management and monitoring of protective equipment (FV20354):** *Ministry of Industry and Trade (MPO)*
2015 - 2019	**Interdisciplinary Research of Wireless Technologies - INWITE (LO1401):** *Ministry of Education, Youth and Sports (MSMT)*
2016 - 2018	**Secure Access-Control for Critical Infrastructures (VI20162018003):** *Ministry of the interior of the Czech Republic (MVCR)*
2014 - 2017	**Secure Systems for Electronic Services User Verification (TA04010476):** *Technology Agency of the Czech Republic (TACR)*
2014 - 2016	**Research into cryptographic primitives for secure authentication and digital identity protection (GP14-25298P):** *Czech Science Foundation (GACR)*

Expert Reviewer

IF Journal	**IEEE Access** *(3.557 Impact Factor), ISSN: 1941-0026*

Certificates

2018	**Post-quantum Cryptography Course:** *Basque center for applied mathematics - bcam, Bilbao, Spain*
2017	**Palo Alto Networks ACE:** *Accredited Configuration Engineer (ACE) Exam*
2017	**Hillstone Networks:** *Hillstone Certified Security Professional*
2013	**Cisco Networking Academy:** *CCNA Exploration: Network Fundamentals*
2013	**Cisco Networking Academy:** *CCNA Exploration: Routing Protocols and Concepts*

Teaching activities

2019 - Present	**ICT Security 1** *- TIC1, Assistant Lecturer, (Laboratory exercise and Lectures, BUT)*
2017 - Present	**ICT Security 1** *- TIC1, Assistant Lecturer, (Laboratory exercise, BUT)*
2016	**Applied Cryptography** *- TAKR, Assistant Lecturer, (Laboratory exercise, BUT)*
2014 - 2016	**Cryptography** *- MKRI, Assistant Lecturer, (Laboratory exercise, BUT)*
2014	**Advanced Data Transmission Technology** *- MVDP, Assistant Lecturer, (Laboratory exercise, BUT)*

Skills

Professional skills	*Cryptography, Privacy-Enhancing Technologies, IoT Security, Smart Cards, Elliptic Curves, Post-Quantum Cryptography, Cyber-Security*
Programming	*C/C++ (Intermediate), C# (Basic), Java (Advanced), MultOS Card (Advanced), Java Card (Advanced), Basic Card (Advanced), Android (Intermediate)*
Languages	*Czech (Native), English (B2), Spanish (A2), Italian (A1)*

Selected publications

IF journals

[1] HAJNÝ, J.; DZURENDA, P.; MALINA, L. Multidevice Authentication with Strong Privacy Protection. WIRELESS COMMUNICATIONS & MOBILE COMPUTING, 2018, vol. 2018, no. 3295148, p. 1-12. ISSN: 1530-8669.

[2] MALINA, L.; DZURENDA, P.; HAJNÝ, J.; MARTINÁSEK, Z. Secure and Efficient Two-factor Zero-knowledge Authentication Solution for Access Control Systems. COMPUTERS & SECURITY, 2018, vol. 77, no. 2018, p. 500-513. ISSN: 0167-4048.

[3] ČLUPEK, V.; ZEMAN, V.; DZURENDA, P. Light-weight Mutual Authentication with Non-repudiation. Radioengineering, 2018, vol. 27, no. 1, p. 143-150. ISSN: 1210-2512.

[4] FUJDIAK, R.; DZURENDA, P.; MLÝNEK, P.; MIŠUREC, J.; ORGOŇ, M.; BEZZATEEV, S. Anomalous Behaviour of Cryptographic Elliptic Curves over Finite Field. Elektronika Ir Elektrotechnika, 2017, vol. 23, no. 5, p. 82-88. ISSN: 1392-1215.

[5] HAJNÝ, J.; DZURENDA, P.; MALINA, L. Attribute- based credentials with cryptographic collusion prevention. Security and Communication Networks, 2015, vol. 8, no. 18, p. 3836-3846. ISSN: 1939-0114.

International conferences

[1] HAJNÝ, J.; DZURENDA, P.; MALINA, L.; RICCI, S. Anonymous Data Collection Scheme from Short Group Signatures. In SECRYPT 2018 Proceedings. 2018. p. 1-10. ISBN: 978-989-758-319-3.

[2] MALINA, L.; DZURENDA, P.; HAJNÝ, J. Evaluation of anonymous digital signatures for privacy-enhancing mobile applications. International Journal of Security and Networks (online), 2018, vol. 13, no. 1, p. 27-41. ISSN: 1747-8405.

[3] DZURENDA, P.; HAJNÝ, J.; MALINA, L.; RICCI, S. Anonymous Credentials with Practical Revocation using Elliptic Curves. In SECRYPT 2017 Proceedings. SCITEPRESS, 2017. p. 534-539. ISBN: 978-989-758-259-2.

[4] DZURENDA, P.; RICCI SARA; HAJNÝ, J.; MALINA, L. Performance Analysis and Comparison of Different Elliptic Curves on Smart Cards. In In 2017 the 15th International Conference on Privacy, Security and Trust (PST). 2017. p. 1-10. ISBN: 978-1-5386-2487-6.

[5] HAJNÝ, J.; DZURENDA, P.; MALINA, L. Privacy-Enhanced Data Collection Scheme for Smart- Metering. In Proceedings of the International Conference on Information Security and Cryptology. Lecture Notes in Computer Science. 2015. p. 1-18. ISSN: 0302-9743.

[6] HAJNÝ, J.; MALINA, L.; DZURENDA, P. Privacy-PAC: Privacy- Enhanced Physical Access Control. In WPES 2014 Proceedings. USA: 2014. p. 1-4. ISBN: 978-1-4503-3148-7.

Reference

doc. Ing. Jan Hajný, Ph.D. — **Head of the Advanced Cybersecurity Group** *at Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Telecommunications in Czech Republic.*
+420 54114 6961, hajny@feec.vutbr.cz

Dr. Jordi Castella-Roca — **Head of the Department** *at Universitat Rovira i Virgili, Department of Computer Engineering and Mathematics in Spain (Catalonia).*
+34 977 558270, jordi.castella@urv.cat