ATTRIBUTE-BASED IDENTITY MANAGEMENT

– Bridging the Cryptographic Design of ABCs with the Real World –

Gergely Alpár

Attribute-Based Identity Management

Bridging the Cryptographic Design of ABCs with the Real World

Gergely Alpár

ISBN: 978-94-6295-052-8
NUR: 980
IPA dissertation series: 2015-01

Typeset using LaTeX 2$_{\varepsilon}$
Cover design by Gergely Alpár
Printed by Proefschriftmaken.nl || Uitgeverij BOXPress

Radboud University

The work in this thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithmics). The author was employed by the Radboud University, Nijmegen. This research is supported by the research program Sentinels (www.sentinels.nl) as project 'Identity Management on Mobile Devices' (10522). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

Attribute-Based Identity Management

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Radboud Universiteit Nijmegen
op gezag van de rector magnificus prof. dr. Th.L.M. Engelen,
volgens besluit van het college van decanen
in het openbaar te verdedigen op donderdag 15 januari 2015
om 10:30 uur precies

door

Gergely Alpár

geboren op 12 april 1974
te Boedapest (Hongarije)

Promotor:

Prof. dr. B.P.F. Jacobs

Copromotor:

Dr. J.-H. Hoepman

Manuscriptcommissie:

Prof. dr. Eric Verheul	
Prof. dr. Jaap Top	University of Groningen
Prof. dr. Josep Domingo-Ferrer	Universitat Rovira i Virgili, Spanje
Prof. dr. Simone Fischer-Hübner	Karlstad University, Zweden
Dr. Thomas Gross	University of Newcastle upon Tyne, VK

Attribute-Based Identity Management

Doctoral Thesis

to obtain the degree of doctor
from Radboud University Nijmegen
on the authority of the Rector Magnificus prof. dr. Th.L.M. Engelen,
according to the decision of the Council of Deans
to be defended in public on Thursday, January 15, 2015
at 10:30 hours

by

Gergely Alpár

born on April 12, 1974
in Budapest (Hungary)

Supervisor:

Prof. dr. B.P.F. Jacobs

Co-supervisor:

Dr. J.-H. Hoepman

Doctoral Thesis Committee:

Prof. dr. Eric Verheul	
Prof. dr. Jaap Top	University of Groningen
Prof. dr. Josep Domingo-Ferrer	Universitat Rovira i Virgili, Spain
Prof. dr. Simone Fischer-Hübner	Karlstad University, Sweden
Dr. Thomas Gross	University of Newcastle upon Tyne, UK

Acknowledgements

It is hard to believe that now I am at the final step of my Ph.D., sitting at home preprocessing my dissertation for printing. It has been an incredible journey with a lot of ups and downs, stresses and excitements, trips and conferences, energising discussions and lonely hours. Of course, many people helped and inspired me. I would like to thank them for their contributions.

First of all, I would like to thank Bogi, my soul mate, partner and wife for everything, for our explorations and for being my universe. She always stood beside me, so I never had to feel by myself.

Bart Jacobs, my promoter, helped in many ways; in fact, he is the most encouraging person I have met. I thank Lejla Batina for giving the best advice, including for suggesting this very project and for being always ready to help me; and I thank Jaap-Henk Hoepman for welcoming me, trusting me, being my daily supervisor, and for all the experiences we had together.

During the writing process of this dissertation I received quite some help. Bart and Jaap-Henk read all the chapters from their early versions on and they gave me a lot of feedback to improve the final result. I thank Eleni and Katja for their 'legal mini-courses' about the immensely complex European regulation. I very much appreciate Peter's, Antonio's and Sam's reviews with regard to certain parts of my thesis. Wouter, thank you for the translation of the 'Samenvatting'; Fabian and Irma, thank you for further improving it. I am also grateful to my reading committee for reading and accepting my thesis; and in particular, to Jaap Top who gave me an extremely useful and exhaustive review.

I was happy to work in the Digital Security group at the Radboud University together with many nice colleagues. In particular, Wouter Lueks, who is being an attentive and thoughtful colleague – I wish we could have talked more; Pim Vullers, who showed me what it takes to be a serious researcher-engineer and, simultaneously, a good person; and Roel Verdult who, together with Lejla Batina, worked really hard on our joint paper in a moment when I really needed their support. I also thank Irma Haerkens for helping in many ways and for always being patient enough while I was talking to her in Dutch. I enjoyed the collaboration within the IRMA team very much, especially, working with Maarten Everts, Roland van Rijswijk-Deij and Ronny Wichers Schreur.

I was lucky enough to closely collaborate with the PI.lab and to get especially inspired by two incredible people: I thank Merel Koning for her refreshing way of thinking and for pouring it on her environment, and Paulan Korenhof for the fascinating discussions – even when she had deadlines – and for her insights that she shares after observing, or rather absorbing, the world.

I would like to thank all my co-authors for the joint adventure that writing a scientific paper means: Bart, Jaap-Henk, Johanneke, Kostas, Lejla, Maarten, Manu, Merel, Paulan, Pedro, Pim, Roel, Roland, Wouter.

Some teachers cannot be thanked enough. Mária Leblanc supported me and made me enthusiastic towards mathematics at an early age. Much later I was fortunate to get lessons from Veronika Hámori, the best maths teacher I have ever known. The professors I learnt number theory and algebra from, Emil Kiss, Róbert Freud and Péter Hermann, had an incredible impact on my future trajectory; they taught me to love these subjects. And finally, Henk van Tilborg, who showed me how to re-discover the beauty of mathematics in cryptography and coding theory, played an important role to drive me to want to work in academia.

It is a pleasure to acknowledge some of my friends who were always there for me, Andrea Borszéki for being so wholehearted and Gyula Altai for being my best pal for a really-really long time. Moreover, my friends from the LF-társaság – Attila, Bori, Sári, Melinda, Zsófi, *et al.* – are an indispensable part of my identity. Finally, Tímea, thank you for the invaluable discussions.

Two important Zsuzsas passed away during this project: my mum, with whom I have a lot in common, and my aunt, who was distant yet influential. I would like to mention here my sisters, Petra and Vera, because they are essential in my life, and my father, Miklós Alpár, because he has always supported me – persistently, ever since my birth.

Gergely Alpár
Nijmegen, November 2014

Abstract

Attribute-based credentials (ABCs) provide a new way to authenticate using selectively disclosed personal attributes, possibly without identification. Smart-card technology has now become sufficiently advanced to implement and deploy ABCs. This thesis focusses on the cryptographic and broader technical challenges of applying ABCs in identity management, both online and offline.

Chapter 2 discusses identity management technologies. We show that many security, privacy and usability issues are present. In our view the main reasons for these problems are the legacy of traditional identity management, having its origin in centralised organisations, and the ubiquity of *ad hoc* solutions devised in the ever expanding digital world. In a more general sense, the main problem seems to stem from the lack of an identity meta-layer and from the overspill of personal data processed by a great number of systems. We put forward recommendations about how to ameliorate this identity crisis and what research directions are ahead of us in this context.

The cryptographic techniques of ABCs provide a novel approach in authentication, one of the main functions within identity management: Personal information can be proven without identification. Chapter 3 provides a description and comparison of the two major ABC technologies, U-Prove and Idemix. In such a system there are two main procedures. In the issuing procedure an issuer (or identity provider) provisions an attribute-based credential to the user, and in the verification procedure the user selectively discloses the necessary attributes from already existing credentials to a service provider (aka a verifier). A smart card is a suitable choice for carrying credentials related to a user since it is secure, personal and stays under the user's control. Furthermore, as recent efficient implementation results show, the smart-card technology is now ready for performing all the necessary computation for ABCs. We call a card with such an ABC implementation an ABC card.

Since attributes are not necessarily identifiable, verification can be completely anonymous. This offers unprecedented privacy for the user. However, the communication between an ABC card and a verifier should be secured to make sure that an adversary cannot eavesdrop on disclosed attributes. Chapter 4 studies this problem. The main challenge lies in the fact that mutual authentication is required for setting up a secure channel, while an ABC card remains anonymous. We introduce therefore a credential that proves validity of an ABC card without revealing any identifying information. Such a credential is issued only to verified cards, and then this validity can be checked by verifiers and used for bootstrapping trust (without identification). We offer two different solutions for establishing a secure channel. Both of them have different properties in terms of efficiency and privacy. In particular, one scheme is more efficient, the other one provides privacy not only for the ABC card but also for the verifier. This latter functionality gives rise to potentially new applications in which the verifier also needs privacy.

Another approach to provide confidentiality for selectively disclosed attributes may be required when a personal device or the infrastructure is not suitable for establishing a secure channel. ABCs can also be implemented on RFID tags, which typically communicate with the tag reader (acting as a verifier) in a simpler way without the possibility to set up a secure channel. Chapter 5 explores this challenge. We give a solution in which the prover (RFID tag) 'wraps' the selective disclosure proof using the public key of a verifier in a way that only this designated verifier can open and retrieve the disclosed attributes along with the corresponding proof. A relatively small change in the verification protocol suffices on the tag's side. However, a modification in the infrastructure is necessary: instead of credential signatures an authentic database is used for valid tag identifiers. The technique offers further cryptographic potential in hiding a statement together with a zero-knowledge proof that it proves; this can be called a 'zero-knowledge proof with statement recovery' referring to the conceptual similarity with a digital signature with message recovery.

Chapter 6 describes a new paradigm in identity management based on attributes. Our starting point is the given cryptographic ABC techniques, the ABC cards and the secure communication required in practice. While designing such an identity management system, one encounters many questions and possible solutions. To provide security, privacy and transparency in attribute-based identity management, we need to introduce new concepts, including credential design, a scheme manager, a card management application. Furthermore, to motivate the need for this technology, we describe several use cases from very simple proofs (like, over 18), through a secure login process, to a privacy-friendly authorisation with multiple attributes (like anonymous membership and age verification) and to the issuance of a new credential based on attributes already present on the ABC card. Finally, in order to set up such a new system, we need to outline a secure card provisioning process that preserves user privacy for the whole life-time of an ABC card. Although there are some open challenges (*e.g.* a privacy-friendly and efficient revocation solution, an implementation of increased security level), attribute-based identity management is becoming practical.

Ultimately, some exciting questions are yet to be answered in the near future. What approach can make secure, privacy-friendly and user-centred attribute-based identity management widely used? Will a top-down or a bottom-up approach succeed? What will the killer application be: a national identity infrastructure, a loyalty system, extended enterprise identity management or some novel business application?

Samenvatting

Attribuutgebaseerde credentials (ABCs) bieden een nieuwe manier van authenticeren door het selectief onthullen van persoonlijke attributen, zonder daarbij noodzakelijkerwijs de gebruiker te identificeren. Chipkaarttechnologie is inmiddels voldoende geavanceerd om ABCs te implementeren en in te zetten. Dit proefschrift concentreert zich op de cryptografische en bredere technische uitdagingen van het toepassen van ABCs in identiteitsbeheer, zowel online als offline.

Hoofdstuk 2 behandelt identiteitsbeheer-technologieën. We laten zien dat beveiligings-, privacy- en bruikbaarheidskwesties hierin de overhand hebben. In onze optiek zijn de hoofdoorzaken hiervan de nalatenschap van traditioneel identiteitsbeheer, ontstaan in centralistische organisaties, en de alomvertegenwoordige ad hoc oplossingen die zijn ontworpen in een zich continue uitbreidende digitale wereld. In meer algemene zin lijken de problemen te liggen in het gebrek aan een identiteitsmetalaag en aan het onnodig opslaan en verwerken van persoonlijke data door een groot aantal systemen. We doen aanbevelingen over hoe deze identiteitscrisis beteugeld kan worden en welke onderzoeksrichtingen er voor ons in het verschiet liggen in deze context.

De cryptografische technieken van ABC bieden een nieuwe aanpak voor authenticatie, een van de hoofdfunctionaliteiten binnen identiteitsbeheer: de correctheid aantonen van persoonlijke informatie zonder identificatie. Hoofdstuk 3 beschrijft en vergelijkt de twee voornaamste ABC technologieën: U-Prove en Idemix. In een dergelijk systeem zijn er twee hoofdprocedures. In de uitgeef procedure voorziet een uitgever (Engels: issuer) de gebruiker van een attribuutgebaseerd credential. In de verificatieprocedure onthult de gebruiker selectief de noodzakelijke attributen van een reeds bestaand credential aan een dienstverlener (of controleur). Een chipkaart is een geschikte drager voor aan een gebruiker gerelateerde credentials omdat het veilig en persoonlijk is en onder de controle van de gebruiker blijft. Bovendien tonen recente efficiënte implementaties aan dat chipkaarttechnologie ondertussen klaar is voor het uitvoeren van alle voor ABCs benodigde berekeningen. We noemen een chipkaart met een dergelijke ABC implementatie een ABC kaart.

Omdat attributen niet noodzakelijkerwijs identificeerbaar zijn, kan de controle ervan volledig anoniem plaats vinden. Dit biedt ongekende privacy voor de gebruiker. Echter, de communicatie tussen een ABC kaart en een controleur moet beveiligd worden om te voorkomen dat een tegenstander (Engels: adversary) de selectief onthulde attributen af kan luisteren. Hoofdstuk 4 bestudeert dit probleem. De primaire uitdaging ligt in het feit dat wederzijdse authenticatie nodig is om een veilig kanaal op te zetten, terwijl de ABC kaart anoniem moet blijven. Daarom introduceren we een apart credential op de ABC kaart waarmee de geldigheid van de kaart kan worden aangetoond zonder (de kaart) te identificeren. Dit credential wordt alleen uitgegeven aan geverifiëerde kaarten binnen een specifiek systeem. Hierna kan deze geldigheid geverifiëerd worden door de controleur en gebruikt worden als fundering voor de daaropvolgende veilige communicatie. We bieden twee verschillende oplossingen voor het opzetten van een veilig kanaal. Beide hebben verschillende eigenschappen in termen van efficiëntie en privacy; in het bijzonder is de ene oplossing efficiënter, terwijl de andere niet alleen privacy biedt voor de ABC kaart, maar ook voor de controleur. Deze laatste functionaliteit maakt nieuwe toepassingen mogelijk waarin ook voor de controleur privacy een vereiste is.

Wanneer een persoonlijk apparaat of de infrastructuur niet geschikt is voor het opzetten van een veilig kanaal is een andere aanpak nodig om de selectief onthulde attributen geheim te houden. ABCs kunnen ook geïmplementeerd worden op RFID-tags. Deze tags communiceren normaal gesproken op een eenvoudigere manier met de taglezer (met de rol van controleur), zonder dat daarbij de mogelijkheid bestaat om een veilig kanaal op te zetten. Hoofdstuk 5 verkent deze uitdaging. We presenteren een oplossing waarin de bewijzer (de RFID-tag) het selectiefonthullingsbewijs zodanig verpakt met behulp van de publieke sleutel van een controleur dat alleen deze aangewezen controleur de onthulde attributen en het bijbehorende bewijs kan terughalen. Een relatief kleine aanpassing in de verificatie procedure volstaat aan de kant van de tag. Echter, er is ook een wijziging in de infrastructuur nodig: in plaats van handtekeningen op credentials moet er gebruikt gemaakt worden van een authentieke database die de nog geldige tagidentificatienummers bevat. Deze techniek is daarnaast cryptografisch interessant omdat het zowel de uitspraak als het zero-knowledge bewijs dat deze uitspraak bewijst verbergt; dit kan een ‘zero-knowledge bewijs met uitspraakterugwinning’ genoemd worden, verwijzend naar het vergelijkbare concept van een ‘digitale handtekening met berichtterugwinning’ (Engels: message recovery).

Hoofdstuk 6 beschrijft een nieuw paradigma in identiteitsbeheer dat is gebaseerd op attributen. Ons beginpunt wordt gevormd door de cryptografische ABC technieken, de ABC kaart en het in de praktijk benodigde veilige communicatiekanaal. Bij het ontwerpen van een dergelijk identiteitsbeheersysteem komt men vele vragen en mogelijke oplossingen tegen. Om veiligheid, privacy en transparantie te bewerkstelligen in attribuutgebaseerd identiteitsbeheer moeten we nieuwe concepten introduceren, waaronder credential ontwerp, een schema beheerder en een kaartbeheerapplicatie. Daarnaast beschrijven we, om de noodzaak van deze technologie te motiveren, een aantal usecases. Deze usecases variëren van zeer eenvoudige bewijzen (zoals ouder dan 18), via een veilige inlogmethode, tot privacyvriendelijke authorisatie met behulp van meerdere attributen (zoals anoniem lidmaatschap en leeftijdsverificatie) en het uitgeven van een nieuw credential gebaseerd op attributen die reeds op de ABC kaart staan. Tot slot moeten we, ten einde een dergelijk nieuw systeem op te zetten, een veilig kaartuitgifteproces schetsen dat de privacy van de gebruiker garandeert gedurende de gehele levensduur van een ABC kaart. Hoewel er nog een aantal uitdagingen zijn (zoals privacyvriendelijke en efficiënte revocatie, en het implementeren van een verhoogd veiligheidsniveau), wordt attribuutgebaseerd identiteitsbeheer in de praktijk haalbaar.

Uiteindelijk moeten een aantal spannende vragen in de nabije toekomst nog beantwoord worden. Welke aanpak kan zorgen voor een brede adoptie van veilig, privacyvriendelijk attribuutgebaseerd identiteitsbeheer waarin de gebruiker centraal staat? Zal een top-down of een bottom-up aanpak succesvol zijn? Wat zal de killer-applicatie worden? Een nationale identitfuriteitsinfrastruuctuur, een loyaliteitssysteem, uitgebreid enterprise identiteitsbeheer of een vernieuwende commerciële toepassing?

Contents

Acknowledgements **vii**

Abstract **ix**

Samenvatting (Dutch summary) **xiii**

Preface **xxiii**

1 Introduction **1**

1.1 Research Question 7

1.2 The Structure of the Thesis 9

2 Identity Crisis: Security, Privacy and Usability Issues in Identity Management **13**

2.1 Defining identity management 14

2.1.1 Federated identity management 17

2.1.2 Related work 18

2.2 Fundamental issues 21

2.2.1 What is identity? 21

2.2.2 Different types of access 24

2.2.3 Trust assumptions 25

2.3 Security Issues 27

2.3.1 The identity provider is a single point of failure 28

2.3.2 The risk of phishing has increased 29

2.3.3 What is the optimal size of a key chain? – *or* – How many identities should a user have? 30

2.3.4 Federations are risky 30

2.4 Privacy issues 31

2.4.1 Linkability across scopes 31

2.4.2 IdP knows all user transactions 33

2.4.3 Proportionality and subsidiarity often violated 33
2.5 Usability issues 34
2.5.1 The 8th Law of Identity: Location Independence 34
2.5.2 “Who am I today?” 35
2.5.3 When complex transactions require multiple credentials . . . 36
2.5.4 User profile management 36
2.6 Conclusions & Recommendations 37

3 ABCs: From Cryptography to Implementation 39
3.1 Introduction 39
3.2 Preliminaries 43
3.2.1 Cryptographic problems 43
3.2.2 Pedersen commitment 46
3.2.3 Credential Signatures 48
3.2.4 Proof of knowledge 50
3.2.5 Blind signature 51
3.3 ABC protocols 52
3.3.1 Issuing a U-Prove credential 53
3.3.2 Issuing an Idemix credential 54
3.3.3 Selective disclosure of a U-Prove credential 56
3.3.4 Selective disclosure of an Idemix credential 57
3.3.5 Unlinkability 58
3.4 Performance of smart-card implementations 59

4 A Secure Channel for ABCs 61
4.1 Introduction 61
4.1.1 Our Contributions 63
4.2 Definition 64
4.3 Secure ABC Channels 66
4.3.1 Security Model 66
4.4 Implicit Card Authentication 70
4.4.1 Security Analysis 72
4.4.2 Practical Considerations 78
4.5 Diffie–Hellman ABC Channel (ABCDH) 79
4.6 Related Techniques 83
4.6.1 Pseudonyms 83
4.6.2 German eID Model 84
4.6.3 Anonymous Authentication 84
4.6.4 General Credential-Based Authentication 84
4.7 Conclusion 85

5 Designated Verifier Proofs 87
5.1 Related work 88
5.2 Preliminaries 89
5.2.1 Cryptographic Preliminaries 89
5.2.2 Selective disclosure revisited 90
5.3 Designated verifier selective disclosure 91
5.3.1 Participants 91
5.3.2 Designated Verifier Identification (DID) 91
5.3.3 Designated verifier selective disclosure (DSD) 93
5.4 The security of DSD 95
5.4.1 Zero-knowledge 95
5.4.2 Security model 96
5.4.3 Security 98
5.4.4 Privacy 100
5.5 Discussion 102

6 An ABC Ecosystem 105
6.1 Introduction 106
6.2 Motivation: Use Cases 109
6.3 Attribute-Based Identity Management 114
6.3.1 Participants and Assumptions 114
6.3.2 Processes 115
6.3.3 Card and its owner – practical considerations 117
6.4 Credential Design 119
6.4.1 An Example Credential Dependency Graph 120
6.4.2 Credential Design Principles 123
6.5 Scheme Manager as the Defining Party 123
6.5.1 Tasks 123
6.5.2 Power 125
6.5.3 Principles 125
6.5.4 An Alternative Ecosystem 126
6.6 Functional Extensions 127
6.6.1 Revocation 127
6.6.2 A Mobile Phone as a Personal Card Reader 128
6.7 Other Attribute-Based Technologies 129
6.7.1 Attribute-Based Credentials 130
6.7.2 Attribute-Based Access Control 131
6.7.3 Related European Projects 131
6.8 Discussion 134

7 Conclusion 135
7.1 Future Research 136
7.2 Discussion 137

Bibliography **141**

Index **153**

Abbreviations **157**

Curriculum vitae **159**

List of Figures

1.1 Dependencies among chapters. 9

2.1 Types of identity management systems. 16
2.2 Federation in identity management. 18
2.3 Yahoo sign-in seal 29

3.1 A schematic view of an Attribute-Based Credential. 41
3.2 Credentials are issued, attributes are shown. 42
3.3 Schnorr identification 50
3.4 U-Prove's issuing protocol 54
3.5 Idemix's issuing protocol. 55
3.6 U-Prove's showing protocol. 57

4.1 Implicit card authentication protocol. 71
4.2 ABC Diffie–Hellman protocol. 80

5.1 Adapted U-Prove selective disclosure protocol. 91
5.2 The designated verifier identification protocol. 92
5.3 The designated verifier selective disclosure protocol. 94
5.4 Security game: designated verifier selective disclosure. 99

6.1 Partial identities (Source: FIDIS). 108
6.2 An example ABC card. 118
6.3 Logged items of an ABC card. 119
6.4 An example credential tree. 121

Preface

“The river is everywhere.”
Hermann Hesse, Siddhartha

I spent the first two years of my PhD project mostly at TNO, an applied scientific research institute in The Netherlands. It was then that I realised how different industrial and academic research are. I have to confess that to come to this revelation was not easy. The two fields use very similar terminology and often definitions are alike. But the goals are far apart. For instance, ‘privacy’ for business participants, and thus for industrial research, means a particular set of regulations that they have to satisfy; it’s basically a hindrance. For the academia, however, ‘privacy’ is one of the main values in a modern society that in the ever digitising world we have to safeguard as much as possible. As a consequence of this difference, research takes very different directions. A ‘privacy tool’, for example, that has to be developed for industry would aim at analysing the current state in terms of privacy at a company, relating it to the regulations and providing concrete guidance how to meet the requirements not yet satisfied. The same term ‘privacy tool’ in the academic world would most likely trigger a research project that helps individuals analyse and manage all the personal data that they use in transactions with other parties.

It is not only the approaches that are different. Industrial research focusses on standards, off-the-shelf products and fast deployment. They generally know little about what is happening in academia. On the other hand, academic research addresses challenging problems arising within their own community. Attribute-based (or anonymous) credentials and their important privacy features, as an example, are essentially unknown to the industrial research community, while academic research considers it, having been around for a decade and a half, almost a commonplace. Even in an industrial or governmental context, identity management could widely use ABCs, but enterprise identity management and related technologies (X509, SAML, LDAP, Kerberos, *etc.*) are easily available and the privacy concerns when such solutions employed are very different, as mentioned above. There are no standards, off-the-shelf implementations, best practices, hence non-academic research does not consider ABCs as a viable choice.

One can argue that applied research, in contrast with pure research, is tailored to fill this gap. But my experience is that the applied side from a pure perspective is still far away from a more theoretical project in industrial research. Therefore, we have to build bridges and bring the two communities closer and closer.

In my view an evidence for this niche in my field is the launch of the Real World Cryptography workshop, started in 2011.[1] Its growing success has shown that there is an increasing interest towards this bridge from both academy as well as companies. The latter can offer problems (and open positions), applied research can find quick solutions and/or develop models with solutions, and ultimately academic research can find new challenges, carry out fundamental research and establish new concepts or even research fields.

The current thesis is a summary of my experiences and contributions during the four years of my PhD project. It intends to build bridges in several different senses: between business and academia, between written protocols and efficient implementations, between a full cryptographic description and an up-and-running system.

These four years were an incredible journey for me. Identity management was the main topic of my research, but in a sense to learn my own identity was at least as hard as the work. After so many years of my childhood, my study period and – unlike for most PhD students – my working years (including my own company and teaching), I was able to spend many hours on working on myself. I had to find my past, present and future, my strengths and weaknesses, and I had to make friends with all of them. Or even better said, I had to build bridges also within myself.

[1] First it was organised by the Newton Institute at Cambridge University in the UK as one of the tribute events to Alan Turing on the 100th anniversary of his birthday; see `http://www.realworldcrypto.com` [last accessed: October 26, 2014]

Chapter 1

Introduction

> "Those who surrender freedom for security will not have, nor do they deserve, either one."
>
> Benjamin Franklin

We live in an information-driven society. Information is power and information has business value. Personal data, an important type of information, has a special role in society on an individual and on a social level. The collection of personal information is ubiquitous.

As a PhD student I had the chance to travel a lot by airplane, so I pick that as an example – with some simplification, *e.g.* no visa or ESTA – to illustrate this ubiquity. Here's how it goes from booking to taking off. After selecting the appropriate flight on an airline's web site, you purchase the ticket online. You have to give explicitly your name and other personal details and possibly your loyalty number to collect points and get some extra service or discount, while implicitly you give further information by means of cookies stored on your computer. The airline company sends you a confirmation e-mail that also serves as a ticket. It contains most details you gave during booking, including the exact dates of your trip. Often you get a reminder a few days before the travelling date about what you have to remember to bring. This message frequently contains ads and promotions about accommodations and other services at the destination. When you arrive at the airport on the day of departure, there is an abundant amount of information flowing about and around you. You are recorded on security cameras. Using your ticket and your passport, you check in your luggage and you receive your boarding pass. You go through the security check, including body scan. When you settle down in the waiting area, you might post a message on Facebook using your mobile to let your friends know how thrilled you are to fly wherever you go, and – since the time moves slowly – you entertain yourself by checking what others posted on your wall. Finally, while you're boarding the plane, you show and possibly let scan your boarding pass and your passport. Here you are, sitting on the plane, ready for the flight.

This simple example shows how intermingled our life has become with information. Personal information, in particular, that we release on purpose, implicitly or perhaps involuntarily. What you post on Facebook you do want your friends to know about; you do that intentionally. By buying a ticket and travelling with it, you know that your personal details are processed by the airline company and the airport staff, even though you may not think about it. There is an underlying contract with terms of conditions that legally drives the business process, and which you have approved by one of your clicks while purchasing the ticket. But there may be several parties that have learnt information about you that you did not intend. Google, if you are using Gmail and thus sending all your booking details through the USA, scans your e-mail for keywords based on which you are provided advertisements. Facebook's employees and the system administrators of your e-mail provider can possibly see what you have written and received as responses. (For instance, they know when you are not at home while travelling.) Moreover, the airline company may use some of the information you gave to improve the profile they store about you. And of course security systems at the airport store your face and your gait.

Privacy in an information-driven society is under threat. Homeland security and criminal investigations argue for surveillance and for increasingly extensive knowledge about individuals. Companies, providing free services on the Internet, need revenue and they build their business models on collecting personal details, constructing individual profiles and selling well-targeted advertisements.

Privacy-enhancing technologies (PETs) aim to design applications that support privacy while preserving the application's most important functions. The design process of a PET includes the collection of security, privacy and functional requirements, and the development of a tool that satisfies those. The tool often involves cryptographic techniques that can achieve sometimes intuitively contradicting goals. Using PETs, security and privacy are not necessarily a trade-off, both can be reached simultaneously.

We acknowledge that the collection and the processing of personal data are often required to achieve certain goals. But is it *always* necessary? And if it is, is it necessary to such an *extent* as we experience now? The *data minimisation* principle states that only those personal details should be collected that are strictly necessary to perform the desired functionality. There are other important principles – such as, transparency, free and informed consent, ensuring data security measures, purpose limitation – in the context of digital privacy. Nonetheless, in this thesis we primarily focus on data minimisation in relation to authentic personal data. As little data should be revealed as possible because personal information that has already been collected and is thus processed by other parties, is difficult to control for multiple reasons. First of all, you have, in general, no or just limited access to data stored at other parties. Second, stored information gets easily duplicated and forwarded, making control even more difficult. Finally, although legal restrictions apply, they are often controversial and they are hard to enforce.

Authentic data is information verifiably asserted by an entity. In person, the authenticity of an assertion is often implicit: If I assert something to you and you see me saying it, then it is clearly asserted by me. However, if you see a document signed by me, the only way for you to verify that it holds my (handwritten) signature is to have prior knowledge of my authentic signature to compare with the one on the document. Also in electronic communication the verification of origin requires some prior knowledge. An electronic signature is bound to a piece of information and an entity's (cryptographic) public key. Anyone who knows the information and the public key should be able to verify the signature; however, only this entity should be able to produce it. So, authenticity of data can be confirmed by verifying the signature on the data through an entity's public key. Of course, this requires the knowledge of this public key and its unique relation to the given entity. This relation is asserted by a trusted party, the Certificate Authority, in the form of a public-key certificate.

What can go wrong when using conventional (non-PET) digital tools? A passport, for instance, is typically issued by a national government administration and meant to be verified at the border control and by the staff at airports. A biometric passport includes a chip carrying the same personal details of its holder as those printed in the document. Therefore, such a passport enables not only physical but also electronic verification. This increases the speed and reliability of authenticating the holder's identity and the stored data, and makes it possible to check efficiently that the present identity document is not on a blacklist of revoked passports. But electronic processing of these data also enables to store information about passengers. There is no user control in place in this respect. All the data extracted from the chip can also be stored at the verifier.[2] If we compare this with traditional, paper passport verification, there is a big difference. Unlike the old process, not only eligibility is checked – *e.g.* to cross a border – but also activities of citizens can be stored, linked and possibly traced. Furthermore, we may not trust verifiers equally. A verifier can be the border guard authorities, but also a whisky shop checking that a customer is older than 18 years old.

One important PET is the technology of Attribute-Based Credentials (ABCs). This technology allows the construction of electronic documents resembling a passport, an airline ticket or a boarding pass. Their authenticity can be confirmed: Each of these credentials are issued by some official entity and, knowing the entity's public key, anyone can verify the authenticity of the information in them.

ABCs can provide an elegant solution without the drawbacks described above. If an electronic passport was implemented as an ABC, the user could reveal different pieces of information from the same credential by using the so-called *selective disclosure* functionality. A border guard may see all attributes from the passport, while a shop assistant only gets to see the customer's date of birth, or even just the fact that the customer is over 18.

Attribute-Based Credentials are not new. The technology was proposed more than a decade ago [Bra00, Ver01, CL01, **?**]. What is new is that smart-card tech-

[2] Why not store it when storage is cheap and the personal information may turn out to be useful in the future? This latter way of thinking can lead to a so-called *function creep* when data are used for other purposes than the ones they were collected for.

nology has improved so much that now their chips can perform all required cryptographic computations. Consequently, we can think of a system in which each user has such a card, an *ABC card*, working like a wallet. It can carry independent credentials, such as an identity card, a bank card and loyalty cards. When using these credentials, one can disclose personal information, or attributes, selectively from them as mentioned above. Many of these attributes are not identifying, in fact they can be completely anonymous, such as '`the residence of city ...`' or '`over 18`'.

Moreover, ABCs preserve favourable properties of the human verification of a paper document. When you are showing your passport at the border control or in a shop, the government agency that issued it does not know that you are using it right now. Also, if you use your passport in two different shops to prove that you are over 18, they cannot link these activities.[3] Similarly, if you used an ABC passport you could satisfy the same requirements digitally: *issuing unlinkability* and *multi-show unlinkability*. Provided that you use only anonymous attributes, the issuer cannot relate the use of your passport to its issuing instance. Neither can shops link your anonymous activities to each other.

These properties have special importance on the Internet when you intend to access some application or service online. To do that, you often need to 'prove' something about yourself. To sign in to a web site, you need to prove that you own a username. This is usually done by typing and sending your password. Another example is when you have to prove that you are over 18 years old. Currently, there are two solutions for that; both of them are poor for different reasons. First, you have to click on a button on the web site "I'm over 18" – this clearly provides no assurance to the service provider. A second solution to prove that you are over 18 is that you have to log in to a system in which you have already registered and supplied evidence about your age. Having identified you, the service provider can check your record in its own database. In this process it learns not only that you are over 18 but also who you are. This clearly contradicts the data minimisation principle and thus it is not privacy friendly.[4] Not only your age but for instance membership would also be desirable to be proven without revealing any other personal information.

Identity management (IDM) has been developed in enterprises and universities. It was natural to store the main attributes of employees and students (identifier, name, position, *etc.*) and control their access rights within the given context. Later IDM faced new challenges when *e.g.* companies merged or the internal IT sys-

[3] We can assume that shop assistants do not care memorising all the names or passport numbers they encounter. Alternatively, we can visualise the idea by always covering all irrelevant fields in a passport when showing the document. In fact, the latter idea was also physically realised by the Qiy Foundation recently: `https://www.qiy.nl/en/respect-privacy/` [last accessed: October 26, 2014].

[4] To see that there exists a more privacy-friendly solution, it is sufficient to say that if you had revealed in two cases *only the fact* that you are over 18, the two activities could not be linked to the same individual. However, this is possible if you have to log in.

tems became required to be available from outside through the Internet. Another challenge arose when universities collaborated with each other and started allowing external students to get access to their own resources when they visited other institutes. This resulted in extending IDM technologies, systems became federated.

Meanwhile users created one account after another for many web sites, including web shops, e-mail services, blogs, government sites and social networks. Each of these accounts required a new pair of a username and a password, which results in turn in deficient password practices [FH07]. A still unresolved problem on the Internet is how users can log in securely and conveniently.

A possible solution is to borrow similar techniques as enterprises apply. Identity management, with a central system, can handle user registration, authentication and user access. But who should be this central authority on the open Internet? This would mean such a power hub that cannot be desirable because of security and privacy reasons. Nevertheless, we see currently such a trend on the web. Huge online companies, including Google, Facebook, Apple and Amazon, provide an increasing number of identity services, often called 'social login'. They authenticate users for third parties. This is convenient for users because they have to remember fewer usernames and passwords. Furthermore, users can enjoy the single sign-on (SSO) functionality on a computer. A user, who has already logged in to one service, can access another one without having to authenticate again. However, users, being constantly traced and profiled, become increasingly dependent on these companies.

Besides these straightforward issues, there are a lot of security, privacy and usability challenges that have to be solved in our information-driven society. After defining concepts in the context of IDM, Chapter 2 discusses these challenges. In the rest of the thesis we attempt to build an identity system in which many of these issues are overcome. To this end, we first recall the essential properties of Attribute-Based Credentials in Chapter 3 in a structured and practical way. Also in this chapter we briefly describe the performance results of the state-of-the-art ABC smart-card implementations of this technology. Then Chapter 4 solves a practical cryptographic problem: How can a secure channel be established between an ABC card and a verifying terminal. This channel has to protect against eavesdropping and it also has to guarantee that only legitimate terminals get access to information on the card. Next, Chapter 5 puts forward a new cryptographic technique, the designated verifier ABC proofs. Surprisingly, this very different approach can also play the role of a secure channel. Finally, using ABC cards, in Chapter 6 we propose ABIdM, a user-centric identity management paradigm with which individuals online and offline can gain access to resources in a privacy-friendly, secure and user-friendly way.

Many results in this thesis stem from the wonderful collaboration within the IRMA[5] project. Some of the contributors with whom I worked a lot were Maarten

[5] `https://www.irmacard.org` [last accessed: October 26, 2014].

Everts, Jaap-Henk Hoepman, Bart Jacobs, Wouter Lueks, Roland van Rijswijk-Deij, Ronny Wichers Schreur and Pim Vullers.

1.1 Research Question

The aim of this research is to gain a better understanding of identity management in closed (enterprise, university) and open (Internet) environments, and to provide a possible new technical approach that outperforms the current technologies in terms of security, privacy, and possibly also usability. This goal can be translated into the following objectives:

- To analyse current IDM technologies and to find weaknesses by challenging them with real-life scenarios in relation to security, privacy and usability, and to formulate essential requirements for improvement. An important consideration is trust and its reduction towards trusted third parties (TTPs) within an identity management system.
- To develop techniques that can satisfy some of those requirements. This stage certainly includes the study of available research results. These results and the technique to be developed are expected to be cryptographic in nature; preserving security while trust requirements are being reduced can typically be solved by cryptography.
- To implement or make a prototype of the most suitable technique on a suitable platform in order to test its feasibility for practical purposes.
- To reconsider the weaknesses and requirements found at an earlier stage of the research and to examine the gaps between the prototype technology and the intended, improved identity management infrastructure.
- To design an infrastructure, based on the prototype, to enhance current identity management technologies in terms of security, privacy, and possibly, usability.

Before we can state the research question, we need to be more specific about the cryptographic approach. We select Attribute-Based Credentials as the scientific starting point for our development process. Below we motivate our choice.

An individual's digital identity is made up of attributes, atomic pieces of personal information. On the one hand, an individual 'gives away'[6] some of his or her attributes to be stored in a particular system. The view of the system about this individual, which is said to be his or her identity there, is the collection of all attributes related to him or her. On the other hand, during authorisation only a small subset of these attributes provides the base for the system to decide whether access is granted or denied to a certain resource. Therefore, it is excessive to retrieve the whole identity instead of only the relevant attributes when an access decision has to be made.

[6] How the personal data is released and collected is irrelevant here.

The ABC technology takes a different view of identity and authorisation. It enables attributes to be issued and stored at the data subject (the individual); moreover, only the relevant and often non-identifying subset of these attributes needs to be shown in the context of a particular verification and authorisation instance. The individual cannot change his or her attribute values; this provides assurance to systems that use attributes for making access decisions. However, there is no way for these systems to relate otherwise anonymous (non-identifying) attributes to digital identities. This duality is the main strength of ABCs: security for the systems, privacy for individuals.

Based on this decision and the objectives above, we establish the main research question of the thesis:

> *How can a new system be built based on attributes that solve the most essential security and privacy problems in the context of digital identity and access management?*

This central question is divided into sub-questions addressed by the chapters of the present thesis:

Chapter 2 What are the main challenges when centralised identity provisioning technologies transform into identity management technologies that need to satisfy complex requirements for the benefit of individuals and service providers on the open Internet? Or from a progressive point of view: What should new approaches be able to satisfy?

Chapter 3 What are the main cryptographic processes in Attribute-Based Credentials and why is the smart card technology a suitable choice for implementing an ABC client? How can the technology be demonstrated to be useful in practice? How do the two main ABC technologies compare?

Chapter 4 How can a secure channel be established with an anonymous ABC card for the issuing and verification protocols? As a secure channel requires authenticity on both sides a sub-question arises: How can an anonymous client be authenticated? Is it possible to achieve for both the user *and* the verifier to remain anonymous?

Chapter 5 How can the ABC verification protocol be adapted to an even more limited infrastructural environment (1. minimal interaction and 2. no certificates), such as Radio-Frequency IDentification (RFID) tags, where a secure channel is not possible or not practical to be established?

Chapter 6 How can ABCs be put in practice as the main building block of a user-centric and privacy-friendly identity management system? Why is it practical and privacy-friendly to use attributes instead of identities? How can a new attribute-based technology help solving the security and privacy problems present in current identity management systems?

1.2 The Structure of the Thesis

Figure 1.1 serves as a reading guide to this thesis. Chapter 2 and Chapter 3 provide the starting points by giving an overview of identity management technologies and by describing Attribute-Based Credential protocols in an accessible way, respectively. Based on these ABC protocols, Chapter 4 and Chapter 5 propose new cryptographic approaches for an ABC client to securely communicate with terminals. Lastly, Chapter 6 provides an infrastructure based on the techniques studied in the previous chapters that addresses many issues from Chapter 2. Although we suggest a linear reading order, we strived to make each chapter comprehensible even without reading the rest of the thesis.

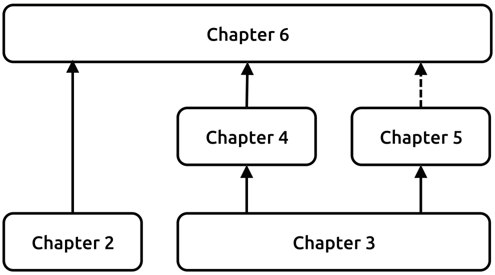


Figure 1.1: Dependencies among chapters.

Chapter 2 explores the state of identity management. First, it studies the main concepts in the field, such as digital identity or trust. Second, it analyses the current technologies from security, privacy and usability aspects by pointing out the main problems still unresolved. Third, it formulates recommendations for possible improvements and suggests open research questions.

> This chapter is an updated version of the paper *"Identity Crisis: Security, Privacy and Usability Issues in Identity Management"* [AHS13] by Jaap-Henk Hoepman, Johanneke Siljee and the author.
>
> *Contribution*
> My contribution in this chapter is to establish an overview of the state-of-the-art in identity management and relate traditional technologies to the contemporary industry solutions. Furthermore, I actively participated in analysing technologies from security and privacy perspectives.

Chapter 3 gives a technical introduction to Attribute-Based Credentials. A conceptual description gives an idea how mathematical constructions can realise ABCs. Using two well-known methods, by Stefan Brands [Bra00] and by Jan Camenisch and Anna Lysyanskaya [CL01, **?**], we demonstrate and compare the main functions and cryptographic protocols. The chapter concludes with a short description about the performance of the most recent smart-card implementations of these two schemes. [MV11, VA13]

This chapter is an updated and widely extended version of the theoretical part of the paper *"Efficient Selective Disclosure on Smart Cards Using Idemix"* [VA13] by Pim Vullers and the author.

Contribution
My contribution in this chapter is providing a conceptual and technical overview of ABCs that partly enabled efficient implementations of the client side of ABC protocols and led ultimately to the IRMA pilot project.

Chapter 4 focusses on an important, anonymous authentication problem in relation to the smart-card implementation of Attribute-Based Credentials. As privacy is a main consideration in the ABC technology, releasing personal information in the form of attributes during a selective disclosure protocol has to be protected. Therefore, a secure channel is required for the transportation of this information. In order to establish such a channel, the anonymous smart card and the verifying terminal have to ensure that they communicate with each other which necessitates mutual authentication. The chapter defines this problem and solves it by proposing two different kinds of cryptographic protocols for setting up a secure channel between an anonymous smart card and a terminal.

This chapter is an updated and extended version of the paper *"A secure channel for attribute-based credentials [short paper]"* [AH13] by Jaap-Henk Hoepman and the author.

Contribution
My contribution in this chapter is the definition of the security model for establishing a secure channel between an anonymous smart card carrying ABCs and an issuing or verification terminal, the design of two cryptographic protocols and the security proofs of these protocols in the given model.

Chapter 5 describes a new notion in the context of Attribute-Based Credentials, the designated verifier proof. Like digital signatures which provide message recovery, this cryptographic technique enables a designated verifier to recover selectively disclosed attributes from a zero-knowledge proof. Any other party learns no information. Interestingly, this technique provides an alternative for a secure channel, discussed in the previous chapter, in case of a simple selective disclosure (*i.e.* containing attributes only from one credential).

> This chapter is an updated version of the paper *"Designated Attribute-Based Proofs for RFID Applications"* [ABL12] by Lejla Batina, Wouter Lueks and the author.
>
> *Contribution*
> My contribution in this chapter is the formulation of the problem and merging techniques of different fields ('Randomized Schnorr' and discrete logarithm representation problem). Furthermore, the final protocol design and the security proofs are joint work with the co-authors.

Chapter 6 proposes an identity infrastructure based on Attribute-Based Credentials. One instance of such an infrastructure is called an ABC ecosystem. We describe the concept of attribute-based identity management, *i.e.* IDM in which all participants manage users' identities by means of their attributes.

> This chapter is an updated and extended version of the paper *"Credential Design in Attribute-Based Identity Management"* [AJ13] by Bart Jacobs and the author. Furthermore, this chapter is based on the collaboration among all participants within the IRMA project.
> *Contribution*
> My contribution in this chapter is the design of an attribute-based identity management system being realised by Attribute-Based Credentials. The new notions and proposals described in the chapter are joint work primarily with Bart Jacobs and secondarily with the whole IRMA team.

Chapter 2

Identity Crisis: Security, Privacy and Usability Issues in Identity Management

Identity management consists of the processes and all underlying technologies for the creation, management and usage of digital identities. In practice, it covers the process of establishing the identity of a remote entity (a human user, a device or a system), managing access to services by that entity, and maintaining identity profiles concerning that entity. Throughout this thesis we mainly focus on human users, although most part of the description can also refer to entities in a more general sense.

Identity management (IDM) is an essential component for the successful development and growth of user-centric Internet services in which users are not only passive observers but active participants of systems. Without general IDM frameworks, public trust in web-based services and applications will deteriorate [Cam05]. Identity theft and privacy violations are an increasing problem (*e.g.* [Fin14]), while keeping track of multiple accounts and passwords is cumbersome and frustrating for users and results in insecure password practices (such as re-using the same account names and passwords at many services) [Cav06]. Secure, reliable and user-friendly IDM is also considered fundamental in establishing trust, for instance in e-commerce applications [Sch11].

Unfortunately, IDM is also a confusing concept, mainly because the different stakeholders involved (users, service providers, enterprises, mobile operators, *etc.*) have different views and requirements. This has resulted in quite a number of different approaches towards providing IDM. Several competing systems exist, most of which are in fact under active development. Their features change from time to time, adding to the confusion surrounding identity management.

The historic development of IDM partly explains how this confusion arose. The scope of identity management used to be a single organisation, such as an enterprise or a university, managing a limited set of services and employees, specific to one application or ICT platform. Currently, this is no longer true. Organisations deliver ICT services to their customers and employees of other organisations as well. This turns IDM into a complex process that has to deal with many applications spanning multiple organisations and multiple contexts, instead of one application within one organisation [Roy13].

The user perspective has also grown in importance. With the increasing presence of organisations on the Internet, and with the creation of a slew of web applications, such as social networks, users start having their own demands for IDM on the web as well (*cf.* [KO02, JZS07]). For users, registering to a large number of services and then managing and remembering different user accounts on such web sites are cumbersome. Entering personal information, like name, address and phone number over and over again with every e-commerce site should be avoided. And finally, the identity management systems (IMSs) that support the use of web applications in a variety of contexts should be privacy friendly by providing control for users over their personal data.

We do not claim that ubiquitous federated identity management (see Section 2.1.1) is universally desirable. In fact, in many cases a small, closed-world setup is preferable from a security point of view. Privacy also cannot be defined in such a universal system as privacy requirements are very much culturally dependent and they change over time [BC11]. Nevertheless, it would often be desirable to have a flexible, well-understood IDM architecture that can be instantiated with specific federation and/or privacy requirements, in which the appropriate security and usability requirements are satisfied as well.

Current systems for identity management do not meet these requirements yet. Apart from the fact that properly implementing an IMS spanning multiple organisations is very complex, there are more fundamental problems already on the conceptual level; see Section 2.2. These systems suffer from several shortcomings that need to be addressed before they can be considered truly secure, privacy friendly and usable. Some of these issues are well known, while others are much less understood. In this chapter we provide a comprehensive account of problems and offer recommendations to resolve or to mitigate them in order to end the current identity crisis.

2.1 Defining identity management

We do not define an ***entity***, but it is understood as a separable unit that can act by itself; for instance, an entity may be a human being, a computer, an organisation, or even a browser session.

In this thesis we define identity within a scope [JWD08] in a technical way as follows.

Definition 2.1. The ***identity*** of an entity within a scope is the set of all characteristics that have been attributed to this entity within that scope. An ***identifier*** uniquely identifies an entity within a specific scope.

We will return to the distinction between identities and identifiers later on in Section 2.2.1. The characteristics of an entity are also called *attributes*; we will elaborate on them in much more detail in Chapter 3 and Chapter 6.

Definition 2.2. ***Identity management*** consists of the processes and all underlying technologies for the creation, management, usage and destruction of digital identities.

In a typical identity management system we can distinguish three parties: users, identity providers (IdPs), and relying parties (RPs)[7]. The user (U) requests a service from the RP that relies on the IdP to provide authentic information about the user. These parties are represented by technical components that cannot be held legally accountable. We therefore use the notion of ***domain*** to represent a legal entity (a natural or juridical person) that is responsible and accountable for the activities of the technical component it operates.

Several types of systems exist to administer identity [PM03, BMH05]. We distinguish ***identity provisioning*** systems (still in widespread use within enterprises) from identity management systems [Cav06]. In the first case, a centralised system manages identities across systems, applications and resources, usually within one organisation or governmental body [CP09]. In the second case, an individual may have multiple IdPs at different organisations and controls which IdP is selected to provide identity information to a RP. Sometimes such schemes are called federated identity management systems. We explain this concept in more detail in Section 2.1.1.

Within the domain of (federated) IDM, we choose to make the distinction between *network-based* identity management and *claims-based* identity management (see Figure 2.1), because their difference in architecture has an impact on the security, privacy and usability issues associated with them (*cf.* [SJ10]).

In a network-based identity management system, the procedure to access a service and to determine the identity and attributes of the visiting user roughly runs as follows. When the user visits the RP, the RP asks the user to authenticate herself at the IdP. The IdP performs this authentication, and if successful, gives the user a *token* that the user forwards to the RP. The RP verifies the token, and if valid, accepts the user as authenticated. To obtain further identity information about the user, the RP contacts the IdP directly, using the token as a pointer to the user profile stored by the IdP. In some cases, the user mediates this exchange of informa-

[7] Relying parties are also known as service providers (SPs) or Verifiers while IdPs as issuers. We use the terminology depending on the context and sometimes in an ad hoc manner.

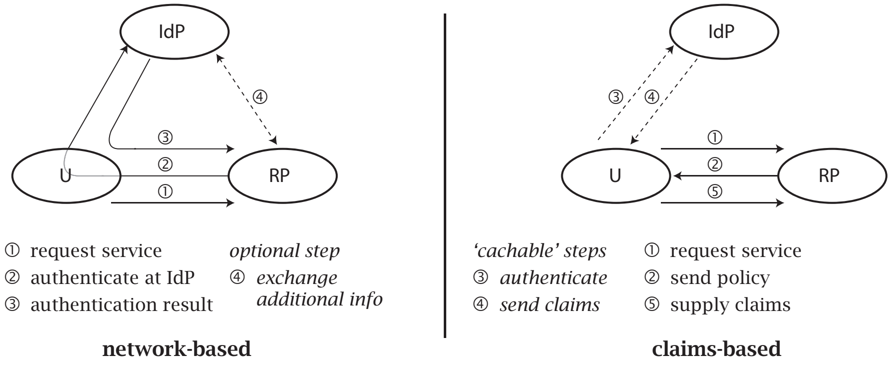


Figure 2.1: Types of identity management systems.

tion between IdP and RP. Examples of network-based IMSs are OpenID[8], Facebook Connect[9], the Liberty Alliance[10] and Shibboleth[11].

In claims-based identity management systems [BBB+11], a RP specifies the user information it needs in order to grant the user access. The user decides if and how it will comply with that request, by obtaining so-called ***claim***s from IdPs. A claim is a statement about a user (similar to an attribute assertion in SAML 2.0[12]), expressed (and signed) by an IdP. To obtain such claims, the user needs to authenticate herself to the IdP, and after receiving the claim from the IdP, the user forwards it to the RP. In claims-based IMSs, user control is a core feature and built in into the system architecture. A natural extension of claims-based IDM is to make the user's client to store certain claims in advance. As a result, the IdP can be excluded from the authorisation process. When we discuss privacy considerations, we will see that off-line IdPs give more privacy for users (see Section 2.4.2).

The crucial difference with network-based IMS is that there is no direct exchange of information between RP and IdP possible, giving the user more control over the exchange of her identity information. Network-based systems, such as Shibboleth, may have the option to exchange attributes through the user (instead of using the so-called back-channel), but in these systems the RP decides whether to use this option. Even though there exist policy tools, such as uApprove[13] for network-based IMSs that allow a user to deny or give consent to releasing her at-

[8] `http://openid.net/developers/specs/` [last accessed: October 26, 2014]

[9] Facebook Connect is a login mechanism based on the OAuth authorisation protocol. `http://developers.facebook.com/blog/post/2008/05/09/announcing-facebook-connect/` [last accessed: October 26, 2014]

[10] Liberty Alliance has become a part of the Kantara Initiative `http://kantarainitiative.org/` [last accessed: October 26, 2014]

[11] `http://shibboleth.internet2.edu/` [last accessed: October 26, 2014]

[12] `https://www.oasis-open.org/standards` and `http://saml.xml.org/saml-specifications` [last accessed: October 26, 2014]

[13] Developed for Shibboleth by SWITCH: `http://www.switch.ch/aai/support/tools/uApprove.html` [last accessed: October 26, 2014]

tributes to a RP, the actual attribute assertion exchange may still take place by the RP and IdP communicating with each other directly. To sum up, in network-based IMSs user control is an add-on and optional, while in claims-based IMSs it is inherent.

Examples of claims-based IMSs are the now obsolete[14] Windows CardSpace [Mal06], which Microsoft is replacing with the more privacy/friendly U-Prove [Bra00, Bra10], and other privacy/friendly concepts from the academic community, such as Identity Mixer, or Idemix [CL01, **?**, IBM12]. In the case of U-Prove and Idemix, claims are in fact anonymous, and are not transferred to the RP directly. Instead, the statement in the claim is proven to the RP in a zero-knowledge fashion. This further protects the user's privacy, because it makes the user unlinkable between two interactions with a relying party. These two technologies are called Attribute-Based Credentials [CKL$^+$11]; see a detailed discussion about them in Chapter 3.

2.1.1 Federated identity management

The concept of federated identity management is sometimes cause for confusion. At times the term is used to describe the collaboration of several RPs to use a single IdP, all within the same domain. In our view, such a setup is the standard form of identity management, in which no real federation takes place. Instead, we define it similarly to Maler and Reed [MR08].

Definition 2.3. ***Federated identity management*** is a setup in which identity is shared across (security) domains.

A Directory Service (DS) in the federated IDM context, which is often called the Where-Are-You-From (WAYF) service, ensures that a user is directed to its own IdP when accessing a service from another domain. Within such a federation, additional agreements can be made for further optimisation, *e.g.* to have a centralised authentication authority. The so-called Circle of Trust (CoT) is the set of domains that belong to one federation. Note that a domain can belong to several federations and therefore can belong to several Circles of Trust. Figure 2.2 shows the differences.

Example federations include national education and research federations based on Shibboleth (*e.g.* eduroam[15], Switzerland's SWITCH[16], UK Access Management Federation[17], Australian's AAF[18], USA's InCommon[19]) and the Kantara Initiative[20].

[14] See announcement: `http://preview.tinyurl.com/MSCardSpace` [last accessed: October 26, 2014]

[15] `https://www.eduroam.org` [last accessed: October 26, 2014]

[16] `http://www.switch.ch` [last accessed: October 26, 2014]

[17] `http://www.ukfederation.org.uk` [last accessed: October 26, 2014]

[18] `http://www.aaf.edu.au` [last accessed: October 26, 2014]

[19] `https://incommon.org` [last accessed: October 26, 2014]

[20] `http://kantarainitiative.org` [last accessed: October 26, 2014]

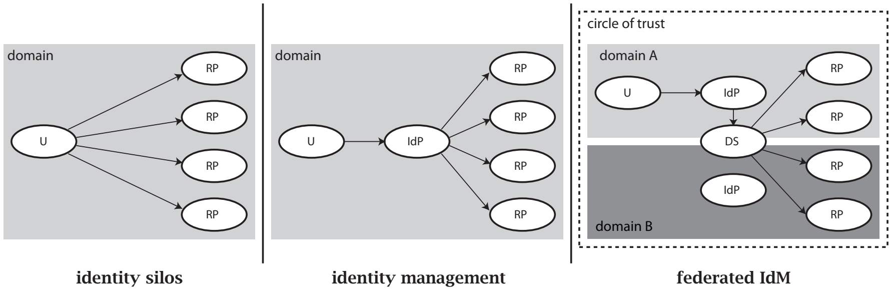


Figure 2.2: Federation in identity management. Unlike in Figure 2.1 where arrows indicated actual information flow, here arrows show logical transfer of identity data.

2.1.2 Related work

Several other studies have stressed the importance of privacy, security and usability of identity management, each focusing on specific issues or looking at the problem from a particular perspective.

Laws of Identity

The seven laws of identity [Cam05] present a compelling set of requirements that a system for IDM needs to obey. The laws were created by Kim Cameron and refined ‘in the Blogosphere’. Below we list all of them:

#1 User Control and Consent: Digital identity systems must only reveal information identifying a user with the user’s consent.

#2 Limited Disclosure for Limited Use: The solution which discloses the least identifying information and best limits its use is the most stable long-term solution.

#3 The Law of Fewest Parties: Digital identity systems must limit disclosure of identifying information to parties having a necessary and justifiable place in a given identity relationship.

#4 Directed Identity: A universal identity metasystem must support both ‘omnidirectional’ identifiers for use by public entities and ‘unidirectional’ identifiers for private entities, thus facilitating discovery while preventing unnecessary release of correlation handles.

#5 Pluralism of Operators and Technologies: A universal identity solution must utilise and enable the interoperation of multiple identity technologies run by multiple identity providers.

#6 Human Integration: A unifying identity metasystem must define the human user as a component integrated through protected and unambiguous human-/machine communications offering protection against identity attacks.

#7 Consistent Experience Across Contexts: A unifying identity metasystem must provide a simple consistent experience while enabling separation of contexts through multiple operators and technologies.

Kim Cameron's Identity Metasystem [CJ07] is an open selector/based digital identity framework, based on the seven laws. CardSpace was Microsoft's instantiation of the Identity Metasystem, but has been discontinued, as discussed earlier, and is being replaced by U-Prove. However, U-Prove is built on another paradigm. While the Identity Metasystem connects individual identity systems to allow seamless interoperation between them, U-Prove is an identity system itself.

Other Related Work

Pfitzmann and Hansen [PH10] collect and develop a consolidated terminology about the fundamental concepts in relation to digital identity and identifiability[21]. Using this terminology as a starting point, Veeningen *et al.* [VDWZ11] develop a formal model and analyse the relations among identity-related properties. Besides the technical problems of identifiability and anonymity, there are many research directions in identity management in relation to security and privacy, such as legal, psychological, security-related, and implementation projects that are discussed in this section.

De Hert [DH08] argues that a legal paradigm shift is necessary in connection with private data. In his view, retailers, governments and other organisations should accept that private information is ultimately owned by the individual who has to be assigned the control over his private data. He also insists that privacy should be a part of the legal framework as well as a fundamental aspect when designing new systems that comprise personal data. The General Data Protection Regulation (GDPR), superseding the current EU Data Protection Directive (DPD), is a huge step in this direction.[22] The new law not only protects citizens' privacy more extensively, but also enforces foreign companies that process data of EU residents to comply with. Besides regulation, Landau *et al.* [LLVGW09] emphasise the importance of *liability incentives* for stakeholders in IMSs.

Dhamija and Dusseault [DD08], considering incentives in a broader technical sense, provide guidelines about how to design a decentralised web identity man-

[21] This effort started in 2000 and after the 34th revised version it stopped with the early death of Prof. Dr. Andreas Pfitzmann in 2010.

[22] DPD or Directive 95/46/EC [Eur95]. The GDPR provides a unified law to regulate the processing of private data within the European Union (EU). At the moment of writing (May 2014) the European Parliament has adopted GDPR's first reading, while the official first reading of the Council is expected to take place possibly in autumn 2014. The Regulation can thus the earliest be adopted in early 2015 and entering into force in 2017, while in all other cases its adoption may take much longer time. [Kos14]

agement system that will take all participants' motivation as well as their capabilities into account. They assert furthermore, that the aspect of usability, allowing users to take appropriate (privacy) decisions, is essential in order to achieve wide acceptance and secure usage of such systems.

However, one of the most challenging research tasks is to build privacy/friendly IMSs with good usability properties. Technical research recommends usable privacy-/enhancing solutions. Jøsang *et al.* [JZS07] propose a scheme that includes a personal authentication device, or PAD, that can support both secure single sign-on (SSO) and protection against phishing attacks. A similar tool, a "smart client", is predicted to gain increasing importance that assists the discovery of appropriate IdPs in complex federated systems [MR08]. Such a device might also assist in a usable solution for *mutual authentication* in which not only users are required to provide credentials, but IdPs and RPs are also authenticated to the users [DD08, JZS07, LLVGW09].

But do people know and care about their digital privacy? According to the survey by Turow *et al.* [TFM05], public awareness about what private information can be stored and resold by RPs is very low and the customers' view is more optimistic than reality. Nevertheless, customers do care about their private data and they are willing to take privacy into consideration in purchasing decisions when information about the privacy statement of the retailer is easily accessible and sufficiently user friendly [TECA07].[23] The same result is described in the study of O'Brien and Torres [OT12] in the context of a social network. The majority of Facebook users change their privacy settings according to their needs but users in general rarely read the "too long and uninteresting" privacy policy. There is a demand for a clearer and shorter policy to improve users' trust.

Although federated identity management solutions are widely employed in corporate and academic environments, many problems still arise. These systems can provide convenient user functions (such as SSO or automated form-filling), however, the single layer of authentication decreases system security [AM07] while it increases the value of user credentials (as it provides access to more resources) [DD08].

Privacy issues emerge in many new technical contexts as well. Pearson [Pea09] collects design guidelines for cloud computing services with proper privacy protection and she describes some open questions (*e.g.* policy enforcement, determination of data processors, constructing privacy design patterns). Another emerging research field is the challenge of building life-long privacy [PBP10] that includes the expansion of solutions to most areas in life and very long-term data security too.

The fundamental technical means for IDM, privacy/enhancing cryptographic tools [CM07] and anonymous credentials [Bra00, CL01] in particular, are available to build privacy/enhancing systems with anonymous but accountable users. In

[23] See further details at the on-line section 'Economics of Privacy' (`http://www.cl.cam.ac.uk/~rja14/econsec.html` [last accessed: October 26, 2014]) by Ross Anderson and the book 'The Economics of Privacy' by Brandimarte and Acquisti [BA12].

the European ABC4Trust[24] (earlier PRIME[25] and PrimeLife[26]) project, building on those technical components, Camenisch *et al.* [CSFH$^+$05] develop elaborate systems that provide more control for users about their personal data by automated negotiation processes. In this thesis we reconsider the building blocks and propose a practical and flexible setup that preserves the main privacy and security features of anonymous credentials; see Chapters 3 and 6.

Eclipse's Higgins[27] – with a practical open-source approach – is a project in progress that is aiming to implement a user/centered identity framework for diverse platforms with a consistent user interface.

The Future of Identity in the Information Society (FIDIS)[28] Network of Excellence provides a wealth of information on the topic, see for instance [BMH05] for a systematic review of identity management systems (see a further discussion in Section 6.7.3 on page 132). Based on their experiences within the FIDIS project, Cameron *et al.* [CPR09] propose a framework for a user/centric, privacy/friendly IDM, with a focus on ensuring interoperability. Their proposal is very much in line with the US National Strategy for Trusted Identities in Cyberspace [Sch11].

2.2 Fundamental issues

There are several fundamental problems with IMSs that arise from the illusive nature of the concepts of identity and trust. Also, too little consideration has been given to the different types of access rights that must be enforced through identity management, as they prove to have an impact on the trust relationships between the parties involved. Because of their fundamental nature, these issues apply to all current models of identity management, and not just the current implementation of such models. We discuss these issues in this section.

In the remainder of the chapter, we study fundamental concepts and security, privacy and usability issues in the context of the current identity management practice.

2.2.1 What is identity?

To begin with, we turn to the concept of identity itself (*cf.* [BM10]). The first thing to note is that identity is *not absolute*. According to Definition 2.1 an identity describes an *entity* within a specific *scope*. For example, you may have one identity within the scope of your job, containing information such as your employee number, and another identity within the scope of your family, containing information on the food you like. Identities are therefore only valid and understandable within

[24] `https://abc4trust.eu/` [last accessed: October 26, 2014]
[25] `https://www.prime-project.eu` [last accessed: October 26, 2014]
[26] `http://primelife.ercim.eu` [last accessed: October 26, 2014]
[27] `http://www.eclipse.org/higgins/` [last accessed: October 26, 2014]
[28] `http://www.fidis.net` [last accessed: October 26, 2014]

a specific scope. If an identity contains many characteristics, it may uniquely identify a particular entity within a scope. However, with only a few (otherwise not identifying) attributes, many entities are likely to match.[29]

It immediately follows that entities have, in general, *multiple* identities. These identities may partly overlap, but can also be mutually inconsistent. The author has for instance blue eyes in all scopes, but goes by different names, nicknames, in different scopes. In extreme cases, people are known to live parallel lives. Sometimes, hardly anybody knows that particular identities in different scopes belong to the same entity.

Identity is *not unique*. Even within a single scope, people may have several different identities. Within the scope of a family a person may not only be a father (to his children) together with all the corresponding characteristics but also a husband (to his wife). Moreover, the identity of an entity is perceived differently by different people, or perceived differently by the same people at different times or in different scopes. Someone may be trusted by one person, but not by another, or only within a certain scope.

To uniquely identify entities, one needs to rely on *identifiers*, not on identities. This distinction between identity and identifier is important. The confusion about these terms is understandable, because in common parlance identity is almost synonymous with personal name, which in turn is understood to be an identifier. Note that also identifiers (such as a user name) are only valid and guaranteed to be unique within a scope.

Digital identities, in the virtual world, can be connected to entities in the real world, but this connection may be loose. For example, computers behind an IP address may be replaced or people may change internet service providers. Likewise, functional roles within companies may look, to external observers, as entities with a particular identity, but different people may actually be assigned to such a role over time.

Identity is also *dynamic*. Assertions about someone's age change when time passes. Your financial situation changes over time, so do your friendships, your convictions and beliefs. IMSs must deal with such changes efficiently, and must avoid keeping old, invalid data.

Identities may exist long after an entity ceases to exist. The *lifetime* of an identity does not correspond to the lifetime of the associated entity. Most of the time identity information is not updated or deleted after it has become inapplicable. This introduces a privacy risk. But sometimes claims about an entity actually need to be kept long after the entity itself disappears. For accountability reasons, Relying Parties store usage information for a period of time, sometimes several years. The situation is reminiscent to the difference in lifetimes between keys and certificates (themselves a possible part of an identity). A certificate needs to be kept long after the key it certifies has expired, to allow parties to verify the signatures made with that key.

[29] The concepts of k-anonymity and ℓ-diversity study these aspects of privacy. [Swe02, MKGV07]

Identity is not only what you want to reveal about yourself, but also what others conclude, believe and find out about you. In fact, most of a person's identity is of this type. Such data may be wrong, become invalid over time, be misrepresented or be misguiding, *etc.*. In other words, an identity does *not necessarily correspond to reality*. Moreover, it shows that an identity *has many owners*:[30] It is not only owned by the entity it describes (aka data subject), but also collected and owned by others. A fine example of this is your medical records being collected by GPs, specialists and other health-care personnel. Health records are often the responsibility of the GP. You may have the right to *view* them, but you do not necessarily have the right to *change* them. This has important privacy ramifications.

Instead of an entity having one single identity containing all characteristics taken from all scopes, it is more natural to view an entity as a collection of multiple identities (a set of sets; *cf.* Figure 6.1 on page 108), each with its own scope. This is also consistent with the idea that privacy guarantees that information about a person does not leak from one scope into another [Nis04].

When scopes merge (*e.g.* if organisations merge) identities may clash.[31] If an entity has an identity in both scopes they may not get merged at all, and as a result the new scope perceives two entities where there is only one. For example, a person may have an account with two different RPs that require the user to use different IdPs. How should this person determine what her identity is in the new scope when the two RPs merge? Or when the two IdPs merge?

The fact that identities may live on long after the entity 'dies' can result in a wealth of personal information stored in many places. This leads to privacy risks for users that are somehow related to this entity. It may also result in IdPs giving out incorrect claims, damaging their reputation. Furthermore, claims (that link some identity information to an identifier) may continue to exist indefinitely, even after the identity information itself is deleted. When the claim of an old identity still exists and a new identity is created with the same identifier, these two may seem to refer to the same entity, while this is not the case.

To summarise, managing identities does not only mean handling new and fixed identities within one scope, but also handling the complex situations of changing identities in changing scopes, and managing the different perceptions of an identity within the same scope.

[30] The discussion is closely related to that about data subjects, data controllers and data processors in the context of the Data Protection Directive [Eur95].

[31] For instance, Microsoft and Facebook are companies that provide identity services in enterprise and social contexts. When these corporations acquire other companies (Skype and WhatsApp, respectively) active in the identity sphere, not only may identities clash but identities belonging to the same entity from different scopes can merge. This raises further (privacy) concerns. Sources: `http://www.bbc.com/news/business-13343600` and `http://newsroom.fb.com/news/2014/02/facebook-to-acquire-whatsapp/` [last accessed: October 26, 2014].

Recommendation A theoretically solid yet practical model should be developed for identity underlying identity management, and IdPs and RPs should make explicit how that model applies to their systems of identity management.

Identity management systems should distinguish between the lifetime of an identity, and the lifetime of claims derived from that identity. They should also provide a way to remove obsolete identities (or part of identities) and to invalidate out-of-date claims.

For instance, to deal with dynamic identity aspects, it would be convenient if a person could get an attribute certificate for the date of birth, which could then be used to prove that the person is older than 18 (without revealing the real age).

2.2.2 Different types of access

Identity management systems are being used to enforce different kinds of access rights. In essence, one can distinguish between *membership* and *ownership* of a resource. These access rights have different risk profiles, and therefore assume different trust relationships between users, identity providers and relying parties. This difference in access rights is not straightforward and it can result in unacceptable risks.

IMSs were first applied in organisations (to centralise access rights management to business applications) and education (to grant students access to the computing facilities, the digital library and the wireless network). In both cases, the identity management systems are used for deciding whether a certain user is a *member* of a group. In the first case it decides whether the user is a member of the group that has access to some business application. In the second case it decides whether the user is a student of a certain university. The resource being controlled is not owned by the user, and any risks or resource damage due to using the identity management system lies completely with the relying party, not the user.

More and more, identity management systems are also being used to enforce *ownership* of a resource. Important examples are online banking, your primary email, and to a lesser extent your blog and social networking accounts. Illegal access to your bank account will hit you with a direct financial loss. Malicious entry to your email account may have indirect consequences, including the ability of impersonating you at other services. Access to your blog and other systems may enable a criminal to 'steal' your identity, which may hurt you in many other ways (*e.g.* your reputation). In this case, the risk of using the identity management systems lies completely with the user.

How does this affect the use of identity management systems? To enforce membership, identity management needs to assume different trust relationships than to enforce ownership. In the first case, the relying party needs *to trust the identity provider* to reliably authenticate its members. In the second case, it is the user that needs to trust the identity provider to reliably authenticate herself. These trust relationships need to be enforced either by technological means, or through mutual agreements like service-level agreements (SLA) with associated penalties. In either case, an identity management system to enforce membership is inherently different from an identity management system to enforce ownership.

Also the risk level associated with using identity management differs. In the case of granting students access to university resources, the damage associated with abuse (and therefore the risk of using identity management systems) is quite low. Except for extreme, denial-of-service cases, the university does not suffer any direct actual loss of non-students having access to the resources. This is the same for any *subscription*-based digital service, such as online music or a newspaper, *etc.* Because the marginal cost of the copy is essentially zero, there is no direct loss if non-members have access as well. The losses incurred by such services are the result of fewer sales.

Granting access to business applications (and the associated data in particular) has a higher risk profile. Not because of loss of revenue, but because the data is often confidential. It could cause enormous financial damage when it becomes public. Similarly, there is a difference in risk level associated with granting access to a bank account and granting access to a blog account.

Recommendation The impact of dealing with different types of resources on IdM deserves further study. For instance, related distinctions that one could make here are on *rivalry* and *durability* of a resource. A *rivalrous* resource cannot be used at the same time by another user, whereas access to a *nonrival* resource does not exclude such access by others (*cf.* common property resources). *Durable* resources do not degrade or get used up, whereas *non-durable* do degrade or can be used only once (*cf.* the difference between 'bits' and 'atoms'). It is interesting to explore the economic literature to see whether even more types of resources and goods can be discerned, and how they influence the trust assumptions in (and the risk of using) identity management.

2.2.3 Trust assumptions

We have been using the ambiguous concept of trust in previous sections, without giving a definition. We will not present a thorough discussion of the notion of trust in this thesis though, but refer to Hardin [Har04], O'Hara [O'H04], Lacochée *et al.* [LCP06] and Jaquet-Chiffelle and Buitelaar [JCB09]. For our exposition the following informal definition based on that of Deakin and Michie [DM97] is sufficient.

Definition 2.4. When an actor ***trust***s another actor, he is willing to assume an open and vulnerable position. He expects the other to refrain from opportunistic behaviour even if there is the possibility to show this behaviour.

In more technical terms, entity A trusts entity B if A relies on the fact that B can break the security or privacy policy of A without A's cooperation or knowledge.

Building trust

Trust can only be built over time. For this, the RP needs to be sure it is talking to the same entity (and the other way around) in different sessions. In order to do so, both parties need to retain information from session to session. Unfortunately, in many of the current identity management systems, the user does not maintain any state. Moreover, the RP is completely relying on the IdP to ensure that the link between different visits of the same user is reliable[32].

The 'proof key' of CardSpace [Mal06] does not solve this issue; it only prevents an adversary from using a security token obtained illegally[33]. The binding is only guaranteed as long as the IdP is honest. If the IdP releases the private proof key or if the IdP itself uses that proof key, the user agent (UA) is no longer involved. The problem could be solved if the UA and the relying party each stored a part of a key pair and verified the link directly without external help.

Trust assumptions are ill understood

By using an IMS, one implicitly agrees to participate in several complex trust relationships between the parties involved. Some of the trust relationships involved in identity management are the side effects of more fundamental security and federation problems, that we will discuss separately later on (see Sections 2.3.4).

The user trusts the IdP not to act on its behalf without his explicit consent. In many systems for identity management, the IdP essentially signs in to a RP, on behalf of a user. It could easily do so, even without the user being present. Clearly, the user does not want the IdP to do this. The general impact of this concern is unclear, though in practice, an IdP that betrays the trust of users may be soon out of business. Additionally, the user expects the IdP not to release personal information unless explicitly asked by the RP and with the permission of the user.

The relying party trusts the federation not to extend the Circle of Trust (without its consent). Depending on the application, a RP may rely on an IdP to provide attributes honestly regarding the user accessing the service. Based on these attributes the RP may decide to grant access to the user or not. A common example is granting access to a wireless campus network to all students, including those that come from other universities. In this application, the IdP will tell the network

[32] This is also a problem with current public-key infrastructures (PKIs), in which the RP also does not keep state and trusts the certificate coming with an authentication instance to ensure a long-term binding between several encounters with the same user over a long period of time.

[33] It works as follows: the proof key pair is generated by the IdP. The IdP sends the private proof key to the user agent (UA), encrypted using the public key of the UA. The public part is also sent to the UA, together with the security token. The entire message is signed by the IdP. Using the private key it received earlier, the UA generates a signature over the combination of token and public proof key, and sends that to the RP. The signature proves to the RP that the UA knows the corresponding private proof key.

whether the user is a student or not. The Circle of Trust (CoT) could be extended when a new university joins the scheme. In this case the federation will delegate the responsibility to classify new members as students to the newly connected IdP. Based on the decision of this new IdP, the network (by necessity) will grant these users access to its network. The decision to grant access to a user is thus in the hands of the new IdP, which may be undesirable (*cf.* Section 2.3.4).

Other trust assumptions involved in IDM. The most basic trust relationship underlying identity management (and this is usually well understood) is the following.

- The RP trusts the IdP to make a particular claim about a particular user.

Although the following trust relationship is equally important and fundamental, this assumption is much less straightforward.

- The user allows the IdP to make a particular claim about her to a particular relying party, and allows the relying party to accept such claims from this IdP.

These trust relationships are also dynamic and context dependent: A user may at some point decide not to use any longer the services of an identity provider, and therefore the trust relationship no longer exists. Moreover, the user may only allow the relying party to accept certain claims from the identity provider within a certain context. For example, if a user only accesses a service from work, or during the day, the relying party should not accept claims about the user during the night, or when it appears the service is accessed from an Internet kiosk.[34]

Every trust assumption is a potential security problem, as the trusted party can break the security policy of the other party. From a security point of view, it is preferable to rely on as few trust assumptions as possible.

Recommendation A better understanding is needed in relation to the trust assumptions among the parties involved in an identity management system. More implicit or explicit trust assumptions should be collected and studied, and it should be determined whether they can be mitigated or avoided by other (*e.g.* technical) means.

2.3 Security Issues

Current identity management frameworks have implemented techniques, methods and policies to securely handle identity information. However, several vulnerabilities remain.

[34] One possibility to capture this notion is via attributes related to the environment (context) which are taken into consideration in the access decision; *cf.* Section 6.7.2.

2.3.1 The identity provider is a single point of failure

Identity management systems require the user *and* the relying party to place a large amount of trust in the IdP (see also Section 2.2.3). A wealth of identity information is stored at IdPs, and users can do nothing but simply trust the IdP to preserve their privacy and properly secure their identity information [DD08]. But still, mistakes can be made and privacy-sensitive information can become public[35]. This makes the identity provider a single point of failure.

Possibly even more worrisome is the fact that currently in most identity management systems the IdP has all information it needs to log in at related RPs as a registered user (see Section 2.2.3). This means that anyone that has access to this information at the IdP can log in as a user to the related RPs; for example, employees of the company hosting the IdP or hackers that break into the IdP systems. Depending on the service, the impostor could order things to make the legitimate user pay money, transfer money from the user's bank account or get insight into personal information, such as the user's electronic health record. The RP has no means to distinguish the impostor from the real user.

This feature can also be (ab)used to turn an IMS into a system for mass surveillance. If the identity provider happens to be the government (and many governments offer IdP services and actively try to extend their use in other domains), then the government has immediate access to all your data stored at services that accept this IdP. Using such an IdP to manage your identity at your bank, your ISP or other relying parties is not recommended in such systems.

The possibly extensive collection of data stored at an IdP can also be used to perpetrate identity fraud. If information about a user stored at the IdP becomes public due to *e.g.* , theft, hacking or implementation flaws, this data can be used to fake an identity when registering for a new service.

The impact of this issue increases as more and more systems get federated and a single IdP is used to access a large number of services. Such an IdP may abuse its powers, maliciously accessing many services, before a user notices. If the RP and the IdP do not properly log authentication requests and access control decisions, both the RP and the IdP may claim that the other party was to blame, and the user will not have any evidence to determine what actually happened.

Recommendation To prevent this issue, it is necessary to put the user in control of information that is released from the IdP [Cam05, BSCGS07]. This should happen not only by policy, but in a technically enforced way in the identity management system. It should not be possible for the IdP to log in to a RP claiming to be a user. The IMS should enforce the requirement that the user controls part of the data necessary to log in to the RP.

[35] Privacy Rights Clearinghouse, "Chronology of Data Breaches" `http://www.privacyrights.org/data-breach` [last accessed: October 26, 2014].

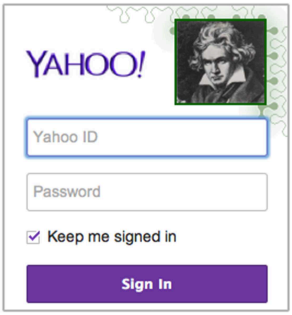


Figure 2.3: Yahoo sign-in seal (with the portrait of Beethoven with green background as the personal seal).

2.3.2 The risk of phishing has increased

Most current identity management systems only provide a way to authenticate the user, but it is not possible for the user to authenticate the IdP or the RP [DD08]. This is a necessity to be able to prevent phishing attacks, in which attackers trick users into revealing identity data and credentials. With identity management becoming more widespread, phishing attacks based on getting IdP login credentials increase as well (*cf.* [Fin14]).

When HTTP redirects are used (as for example in OpenID, in OAuth or in Facebook Connect) phishing attacks are even easier to launch[36]. It is as simple as creating an illegitimate but attractive website that redirects to a false copy of the IdP to capture the user's credentials.

An example countermeasure to phishing attacks using fake IdP websites is the Yahoo sign-in seal;[37] see an example in Figure 2.3. This is a personalised image or a short text phrase that appears each time a user logs in to a Yahoo web page from the same computer on which the seal was created. The presence of the seal enables the user to distinguish the real Yahoo sign-in page from a false page. This solution only works on a single device, as Yahoo identifies it by storing tags in multiple places on the computer[38].

Recommendation To prevent phishing attacks, it is very important that users can (and will) authenticate the RP and the IdP. Mutual authentication therefore needs to be incorporated in identity management systems, in such a way that the user is not required to install special software or to use one and the same computer all the time (as is the case with Yahoo sign-in seal or Microsoft CardSpace). Furthermore,

[36] "Beginner's guide to OpenID phishing", `http://preview.tinyurl.com/openid-phishing`, [last accessed: October 26, 2014].

[37] `https://protect.login.yahoo.com/` [last accessed: October 26, 2014].

[38] "How does Yahoo Sign In Seal Work?" by Gabe Wishnie, `http://preview.tinyurl.com/mvxu6kl`, [last accessed: October 26, 2014]. The

authentication of the IdP and RP by the user should be more user friendly than checking their SSL certificate manually. Apparently, there is no single, usable and secure fix to prevent phishing in all cases.

2.3.3 What is the optimal size of a key chain? – *or* – How many identities should a user have?

One of the main advantages of identity management for end-users is single sign-on: not having to remember all those user names and passwords, except for the log-in token for the IdP. From this perspective, it would be great to have just one IdP: only one user name/password (or another authentication token) and that's it.

Obviously, this is not feasible. Not only because users may not trust that single IdP to have access to all their services (see Section 2.3.1). Even if users *do* trust a single IdP for that, using only one IdP means that if that IdP is compromised, all identity data is compromised immediately as well. It is therefore advisable for users to distribute their identity information over multiple IdPs. Furthermore, different RPs typically require different IdPs. Financial institutions, for example, have other requirements and preferences than car rental agencies with respect to an IdP. The first may want to set up their own IdP to be able to control the security of authentication, while the latter is satisfied with using a third-party IdP. Can we then settle for one IdP for personal use, one for work, and one for each financial institution? This seems to be a workable yet quite arbitrary subdivision.

The question is: How many identity providers does a user need? What is the best compartmentalisation of the digital identity mess? We need to understand the advantages and risks of using a certain amount and distribution of IdPs and federations, in terms of security, usability and business.

Recommendation To be able to determine which and how many 'identities' are optimal, a model that captures these relevant aspects needs to be developed. To our knowledge such a model does not exist yet.

2.3.4 Federations are risky

In cross-domain settings, one organisation may assign roles to certain individuals, while another organisation assigns access rights to roles. This is typically done in federation settings: One university classifies certain people as staff or students of that university, while other RPs rely on that classification to mediate access to resources like the library, classes or the student restaurant.

This gives rise to a compliance defect [Ell99]: The IdP may interpret the semantics of the role (*e.g.* when someone is classified as a student) differently from the RP, which leads to a situation where a person gets access to a service that he or she is not supposed to access. The reverse (being denied access) is a problem as well.

The issue described above is an instance of a more general one. Traditionally, access to a resource or service is mediated through a ‘reference monitor’ (*cf.* an access matrix [Lam74]). In a federated identity management system, this reference monitor is in a sense distributed over several parties. The underlying question is how to do this ‘split’. In the simplest case, this question surfaces as the question: Where to keep the access rights?

A separate issue is the control over the Circle of Trust (CoT). By establishing a federation among several IdPs, the CoT is similarly extended. RPs connected to a certain IdP may have limited control over this, and therefore have limited control over the risks that they are exposed to because of the extension of the CoT; see also Section 2.2.3.

Recommendation When implementing or joining an identity federation, RPs need to carefully consider where to keep and maintain the access rights. Moreover, they need to judge the consequences when the Circle of Trust is extended without their knowledge or consent.

2.4 Privacy issues

Identity management systems are used to facilitate millions of user transactions on the Internet each day. They mediate between users and relying parties, handle a lot of personal information, and often register who does business with whom. This has obvious privacy consequences. We discuss these issues in detail below.

2.4.1 Linkability across scopes

Like AdSense[39] and DoubleClick[40], identity management systems have the potential to track a single user over all the websites she visits.[41]

To maintain privacy, it should be possible for users to be anonymous or use pseudonyms at RPs, and to choose IdPs that do not link all user transactions at all RPs together, and so do not keep records of everything each user has been doing. Many identity management systems implement at least part of these solutions, which is why the UK Information Commissioner has recognised federated identity management as a PET.[42]

However, not all identity management systems do. An example is DigiD, the Dutch national authentication provider that enables authentication of Dutch cit-

[39] `https://www.google.com/adsense/` [last accessed: October 26, 2014].

[40] `http://www.google.com/doubleclick/` [last accessed: October 26, 2014].

[41] Google acquired AdSense (2003) and DoubleClick (2007-2008). Google later changed its privacy policies (2012) by unifying them; and by doing this, a user’s all activities can be used for advertisements throughout Google’s services.

[42] See the document here: `http://preview.tinyurl.com/InfComPET` [last accessed: October 26, 2014].

izens when communicating with Dutch government institutions. DigiD uses the BSN (Burgerservicenummer, Citizen Service Number) to identify a user: After authentication, DigiD sends the BSN to the RP. This number uniquely identifies each user, and does not allow for anonymity or pseudonymity at all. As indicated in [MVS08], expanding the scope of DigiD to incorporate not only governmental organisations but also the private sector has many advantages. However, as already mentioned, in its current form DigiD does not allow for pseudonymity as the BSN is always used as identifier. Such an extension of use of DigiD for the private sector is not acceptable as it violates the user's privacy when all RPs always receive the BSN as user identifier. This enables them to track people and possibly learn who the user is.

Another example stems from the need to retain user permissions at RPs when a user moves from one home organisation (and thus IdP) to another. For example, the educational sector is talking about such an identifier to track students throughout their entire school life. Federations often solve this by implementing one static user attribute (often a pseudonym identifier) that a user can 'bring' to another IdP. This 'feature' severely limits the privacy of users, as the static attribute links all user actions at all its previous and current IdPs and one (but often many) RPs.

This also involves a paradigm shift from identity management relying solely on identifiers. It has become standard practice to require a user to identify herself before granting access to a service. In many cases this is unnecessary. For example, in order to be allowed to buy alcohol, someone only needs to prove that he or she is over 16 years old (or 18 or 21, depending on local laws). Such 'attribute' or 'credential' based forms of privacy-friendly identity management do exist in theory but are rarely applied in practice. (As an attempt to change this, we describe a practical approach called attribute-based identity management in Chapter 6.)

Recommendation Whenever using identity management systems, one should always try to implement maximum anonymity and pseudonymity where possible. A solution for expanding DigiD to the private sector, for example, is to use pseudonyms that are based on DigiD as identifiers. Each user will have a different pseudonym with each RP, and no pseudonym should leak any information about the underlying BSN. A possible method for generating pseudonyms is making a hash of the BSN together with (the domain name of) the RP.[43] Alternatively, if it needs to be possible to trace the pseudonym back to the original BSN, various encryption methods can be used [MVS08]. Furthermore, a privacy-friendly method for retaining access rights at RPs when changing IdP is necessary.

[43] This is a similar approach to the construction of service-specific pseudonyms in the German eID system; see *e.g.* [PWVT12] and Section 6.7.3.

2.4.2 IdP knows all user transactions

In current identity management systems the IdP is involved each time a user requests access at a RP. Therefore, the IdP can keep track of all these user actions (although sometimes the specific RP involved may be kept hidden from the IdP). In most systems the user is not even involved in the exchange of her identity information between the IdP and the RP. But even in a claims-based identity management system, such as CardSpace, where the user needs to give consent before identity information is transferred, even though the IdP does not need to know exactly who the RP is, the IdP often needs to generate the assertion online and therefore knows about all user transactions.

It seems PKI is the solution to this issue. Here a Certificate Authority (CA) identifies and authenticates a user only once, and then certifies the user's public key. The user can then authenticate herself to a RP by signing data with her private key, which the RP can verify using the corresponding public key. In this case the CA is not directly involved in the user authentication by the RP, but is still the trusted third party. There are two important drawbacks of this solution. First, the user always needs to have her public-key certificate available when logging in, and thus PKI-based identity management often violates the 8th Law of Identity about location independence (see Section 2.5.1). This problem arose also in Identity Selectors as used in CardSpace, where all identity selection solutions were hardware-specific, OS-specific or even browser-specific. Second, the same user key (along its certificate) is used at each RP, making the user's transactions traceable.

Recommendation We need to develop an identity management system that does not require IdPs to see all user transactions, without violating the 8th Law of Identity. This apparent paradox may be solvable by relying on personal hardware (like tokens, smart cards), or by developing mobile and cloud-based identity management concepts. A possible solution is described in Chapter 6 which relies on smart cards and attribute-based credentials [Bra10, IBM12, CKL$^+$11, VA13].

2.4.3 Proportionality and subsidiarity often violated

In the EU, most of the data protection or privacy laws are based on the principles of proportionality and subsidiarity. Proportionality stipulates that the amount of personal data being collected is proportional to the goal for which it is being collected. Subsidiarity demands that the same goal cannot be achieved in a more privacy-friendly way.

This is implemented in the EU's Data Protection Directive [Eur95] (see also Section 2.1.2) as Articles 6 and 7. Article 6 can be summarised with the notion of 'data minimisation'. It entails a set of requirements such as purpose specification, use limitation, accuracy and completeness of the data, and deletion and anonymisation of the data as soon as they are no longer needed for the purpose that led to their collection. Article 7 lists all the legal grounds, including the consent of the data subject, based on which one is allowed to process data.

Often, websites and services violate these principles. One does not need, for instance, to identify someone to determine his age. Subscriptions for a service can certainly be handled anonymously. An online newspaper does not need to know *who* accesses the system, all it needs to know is whether that person is *entitled* to read the news online (*cf.* [Ell99]).

Recommendation Less is more. RPs should be precise about the personal information required to offer a service, and should not ask for more information 'just because they can'. An important approach is to consider anonymous ways to offer the same service.

2.5 Usability issues

Usability plays an essential role in security and in the protection of private data as many attacks, such as phishing and social engineering, target users instead of computers [And08, Chapter 2]. Besides these issues, identity management encounters further difficulties. We discuss those challenges in this section.

2.5.1 The 8th Law of Identity: Location Independence

The seven laws of identity [Cam05], as discussed in Section 2.1.2, present a compelling set of requirements an identity management system must satisfy. However, one important usability aspect is missing: location independence. A number of current identity management systems depend on persistent data stored locally at the user's machine. Instead, a user should be able to access a RP using the identity management system not only from her own PC, but also from her laptop at work, her smart phone and a computer at a cybercafé *e.g.* in Hong Kong. This location independence should be accomplished in a secure and privacy-friendly manner.

Note that although some identity management systems fail to implement location independent access, many other identity management systems do provide it. However, they have other privacy and security issues that we have discussed, such as IdPs that know all user transactions (see Section 2.4.2).

We therefore propose the 8th Law of Identity:

Definition 2.5 (***Location Independence***)**.** The identity system must allow a user to create, manage and use her identity independently of her current location and current device in use.

Recommendation Identity management systems should not rely on any persistent data stored locally at the user's machine. However, implementing the 8th Law of Identity should not lead to the security and privacy issues that current location independent identity management systems have.

We foresee two emerging technologies that can support identity management with regard to this law: mobile communication and cloud computing. A mobile phone can for instance assist a user in authentication as a second communication channel or it can act as a trusted computing and storage device. Data required to perform authentication can, on the other hand, be distributed to locations in the 'cloud' that are securely and privately accessible for the user.

Hardware tokens that can easily be carried around can be used to achieve higher levels of security at authentication than are currently possible with cloud computing or mobile phones; but possibly with less user convenience.

2.5.2 "Who am I today?"

As discussed in section 2.2.1, users may have several identities, even within a single scope. This distinction in identities manifests itself when people have several different responsibilities, or, in other words, may have several different 'roles'. Examples may help to clarify this issue.

When signing a document, a notary can choose to sign this as a notary or as a private person. The distinction is legally significant. The CFO of a company may use an electronic banking system either to enter a personal or a business transaction. An ICT system administrator may sign in to a system either as 'root' (which allows him to run OS-level applications and scripts) or as an ordinary system user (that allows him to only execute end-user applications).

We see that users can have different roles that allow them to do different things within a certain service. Furthermore, the impact of user actions depends on their role: A signature of an accountant or a notary represents more legal value.

Current identity management systems do not make it easy for users to manage such different roles. Basically, users are forced to maintain and manage several identifiers to separate these roles. But this may lead to confusion. For instance, if a user has previously signed in at its IdP using a particular identity, and the user and the service support SSO, the user may automatically be signed in using this same identity when accessing a different service some time later. This is potentially dangerous: If the CFO has signed in as CFO earlier, he may not want to execute a personal transaction while still being signed in as CFO.

For many current identity management systems these very common usage scenarios pose a problem. There is no way to indicate as which role or which identity a user wants to access a particular service, especially if she has accessed that system in both capacities before. One of those identities may be selected automatically (in a single-sign-on context), most likely without the user knowing why or how to change it.

Recommendation Identity management systems should provide a way for users to see and select their identity with which they 'sign in' even if explicitly signing in is not asked for, because the user has already authenticated with an IdP that is recognised by the RP. Asking users each time which role (at which IdP) they want to use is cumbersome for the user, and therefore not a good solution to this issue. So, alternative approaches need to be investigated.

2.5.3 When complex transactions require multiple credentials

A special case of the previous issue is that of transactions that require the cooperation of many services, possibly of multiple RPs. This is for example the case in service-oriented architectures (SOAs), where one application consists of multiple software services. The problem arises if the user needs to present credentials for more than one service, and the credentials depend on the role the user assumes. The user needs to have all the credentials required to perform the transaction, but can only present them if she has logged in using the right role. Also in this situation the user has no means to select her role or identity for a particular session.

Recommendation Clearly, this is part of a more general problem of implementing chains of transactions, in which identity management plays only a partial role. But perhaps identity management systems could provide a way to automatically determine the full set of required credentials and the minimal role the user can assume that covers those credentials.

2.5.4 User profile management

When a user accesses a service, this often involves the processing of personal information. Some of that information may be stored at the IdP, while other information is stored at the RP because it is service specific and the RP needs the information for *e.g.* marketing research, or because the RP does not trust the IdP to store the required information. More often than not, many RPs store the same information for a particular user. How can such a scattered profile be managed and be kept up to date by a user? Should a user always be allowed to update such information (consider for example the counter-example of medical records)? The question is whether identity management systems will be able to simplify user profile management both for the end users as well as the RPs and IdPs.

Recommendation This issue is resolved by following the emerging trend towards the convergence of profile, identity and authorisation (or access) management into a single system for identity management. Such a system would also be beneficial for users, as it allows them to manage personalisation of many different services in a central location. This way, changing a single setting once will change the behaviour of all services consistently. This enables a ubiquitous personal experience across many different services. We see a similar trend in social networks, such as Facebook, LinkedIn and Google. These companies stimulate their users to manage their profiles in their services and possibly, share those data with other systems.

The emerging trend of personal data lockers or personal data brokers could be used to enable users to manage their identity information and keep it up to date. The personal data locker is a cloud-based service where users maintain their own data and control who has access to it. This way a user can give a RP access to personal data attributes independently of an IdP.

Note, however, that although identity management is positioned as the solution for the cumbersome maintenance of identity information, the nature of certain businesses (see Section 2.3.4) and the nature of identity itself (as discussed in Section 2.2.1) limit the implementation of management of identity information across organisational domains.

2.6 Conclusions & Recommendations

Identity management not only comprises identification and authentication, but also access management and user profile management. Stakeholders such as end-users and relying parties require identity management systems to be able to span multiple organisations, to be user friendly, privacy friendly and secure. Current systems for identity management are not able to accomplish this.

As we have seen in this chapter, security, privacy and usability are not adequately addressed in current identity management solutions. This renders current systems for federated identity management inapplicable for 'high-value' services, such as electronic banking, that consequently remain to rely on their own home-grown systems for access control.

Federation, as well as the more fundamental concept of identity, and its consequences regarding scope, responsibility and trust, is still not understood. More fundamentally, the term federation is used confusingly within the field causing further uncertainty.

The issues of identity management systems presented in the chapter cause the current *identity crisis*. In order to resolve the identity crisis, we recommend to further investigate the following main observations made in this chapter.

- A proper and practice-oriented model for identity underlying identity management should be developed, and IdPs and RPs should make explicit how that model applies to their systems of identity management.

- Building on that model, the trust relationships between the parties involved in the identity management system should be investigated and formalised.

- To prevent phishing attacks, it is very important that users can (and will) authenticate the RP and the IdP. Mutual authentication therefore needs to be incorporated in identity management systems in such a way that the user is not required to install special software or to use one and the same computer all the time.

- To enhance user privacy we recommend that users can remain anonymous or use pseudonyms at RPs, and to have IdPs that do not link all user transactions at all RPs together. Although identity management systems already implement at least partly these solutions, not all do so. We need an identity management system that does not allow IdPs to see all user transactions, without violating the 8th Law of Location Independence (which states that identity management systems should not rely on any persistent data stored locally at the user's machine).
- Identity management systems should provide a way for users to see and select their identity with which they 'sign in' even if explicitly signing in is not asked for.
- Identity management systems should provide a way to automatically determine the full set of required credentials for a certain service, and the minimal role the user can assume that covers those credentials.
- Finally, we need identity management systems that put the user back into control and that support the user in maintaining a user profile that can be used (in a controlled manner) by business from several organisational domains.

Most of these recommendations are not trivial, and to implement them requires a substantial research, development and standardisation effort. Moreover, to resolve the identity crisis, stakeholders need to work on this together. We believe the growing need for a proper, well-founded, identity management solution is worth the effort.

Chapter 3

ABCs: From Cryptography to Implementation

> "[O]ften the most important contribution a scientist can make is to discover a new way of seeing old theories or facts"
>
> Richard Dawkins [Daw06]

In the previous chapter we studied identity management (IDM) and introduced the main participants: identity providers, users and service providers (or relying parties). An important functionality in IDM is authentication, that is, the process in which the user's identity is verified. Authentication in a broader sense means the verification that some identity information is true about a user.

In this chapter we focus on Attribute-Based Credentials (ABCs) by discussing the main cryptographic building blocks, protocols and implementations.[44] Our step-by-step description shows the abstract cryptographic goals and how they are realised in two major technologies (U-Prove and Idemix, see below more details). This parallel discussion makes it easy to see the similarities and differences between them. The abstraction and the gradual approach have also helped our university group to create efficient smart-card implementations of these technologies and later to put ABCs in practice in an experimental, pilot infrastructure (*cf.* Chapter 6).

3.1 Introduction

ABCs enable a special type of authentication. First, the ABC authentication does not necessarily identify the user, only provides authentic assertions about the user. Second, despite the reduced amount of information, the service provider can make an established access decision. Third, the whole process is user centric. All data is under the user's control that needs for the authentication, so she does not need to involve the identity provider while authenticating at a service provider. Fourth, authentication instances are not only possibly non-identifying (as mentioned above) but also unlinkable. This improves the user's privacy. Clearly, the notion of authentication here is different from usual mechanisms, such as a login process to a website with a username and the corresponding password.

[44] ABCs can be described in various ways. We choose an intuitive approach.

The Attribute-Based Credential (ABC) technology uses also different names for the main roles. An identity provider in the context of ABCs is called an *issuer*. Issuers verify identity information of the user and create credentials for her. The way issuers perform identity verification is mostly out of scope here. Using these credentials, the user can authenticate at a service provider, called a *verifier*.

Identity information can be divided into attributes, mostly used for user authentication at verifiers in an ABCs system.

Definition 3.1. An ***attribute*** is any indivisible piece of personal information that can be described by a bit-string, such as an identifier, a qualification or a property of an entity.

An attribute formally can be described as its *name* and its *value*[45]. The attribute value is a bit-string, but cryptographically we work with integers; how the encoding happens is irrelevant as long as it can be carried out by all participants and is collision-free. Mostly, we say ‘attributes’ referring to both attribute names and attribute values, as long as it is clear from the context. Although in this chapter it does not matter what kinds of entities (individuals, devices, sessions, *etc.*) are described by the attributes, our primary focus in the thesis is attributes related to human users.

Informally, an Attribute-Based Credential is a cryptographic container of attributes that provides security assurance for all participants in the system. Most importantly, the user cannot forge credentials or change the attribute values. Furthermore, to guarantee non-transferability and to bind a credential to its carrier device, the following method is applied. One of the attributes is in fact a *secret key* that is only known to the user’s device. Since this key never leaves the device but is required when the user authenticates to a verifier, credentials cannot be transferred. Relying on these features, a verifier can be convinced that a user’s attributes in an authentication process come from the issuer, and as long as the verifier trusts the issuer with respect to these attributes, the attributes actually hold for the user.

An ABC can be described by the following components:

- the credential’s name;
- the secret key;
- the pairs of attribute names and values;
- the issuer’s identity; and
- the issuer’s signature.

Figure 3.1 shows a simplified view of an Attribute-Based Credential. Throughout this thesis we often describe a credential only by its name and the attribute names. The issuer’s identity is mostly implicit though in some examples we do make that explicit too. A user can have several credentials on her device, *e.g.* , ‘citizen identity’, ‘driving license’, ‘airline loyalty’; see several examples in Chapter 6.

[45] We do not distinguish various *types* of attributes

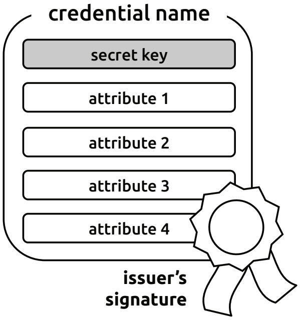


Figure 3.1: A schematic view of an Attribute-Based Credential.

Attributes are stored on the user's device within ABCs. This device has to take part in two important types of interactive, cryptographic protocols: issuing and verification. *Issuing* is a protocol between the issuer and the user's device. As a result, a new credential gets stored on the device. This process is comparable to steps 3 and 4 in claims-based IDM in Figure 2.1 on page 16. *Verification* is an authentication process during which the user reveals and the verifier receives a subset of the attributes from the user's device; *cf.* steps 1, 2 and 3 in the same figure.

ABCs are never shown in their entirety but only some components – and most importantly attribute values – are disclosed from them. Conceptually, any set of attributes on the user's device can be revealed. This is cryptographically realised by multiple *selective disclosure* (SD) protocols.[46] In fact, as many SD protocols are required within a verification as the number of credentials containing the revealed attributes. A SD protocol involves two parts: The user's device discloses a subset of attributes from a particular credential and it proves to the verifier that these attributes are indeed in the credential. Figure 3.2 shows the attribute flow in an ABC system from the perspective of the credential carrier device: issuing and showing. (This latter function is the main building block of the verification protocol.) Abstractly, a verifier receives the following pieces of information as a result of a selective disclosure protocol:

- the credential's name;
- the disclosed pairs of attribute names and values;
- the issuer's identity; and
- a proof that the user has a valid credential of this name containing the disclosed attributes.

[46] The technique to compose several selective disclosure proofs into one proof of knowledge is briefly discussed in Section 4.6 on page 83.

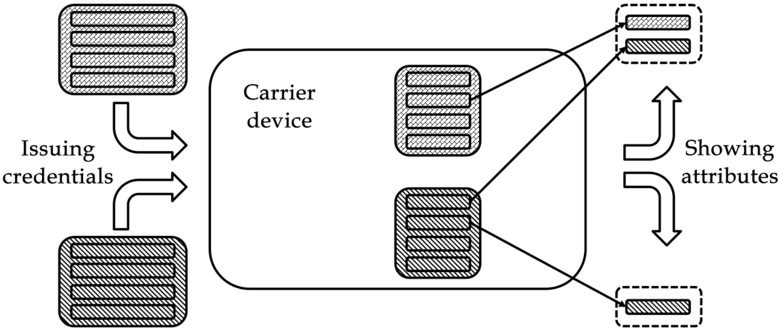


Figure 3.2: '*Credentials are issued, attributes are shown.*' (Signatures are illustrated as watermark to show how a credential signature effects a proof with disclosed attributes.)

For instance, a possible result is the following proven statement: "My 'residence permit' credential is signed by the GBA (the Dutch municipal administration) and I disclose two attributes: '`gender`'='`male`' and the '`city of residence`'='`Nijmegen, The Netherlands`'." Note that this statement is not identifying in general because there are lots of people to whom this assertion holds.

It is clear that unlike many IDM technologies, ABCs enable the verifier to receive authentic data from the issuer without directly communicating with it. Even more, the issuer does not need to be involved in the transaction whatsoever.

In this thesis we consider two major ABC technologies, Microsoft's U-Prove [Bra10] and IBM's Idemix [IBM12]. U-Prove is based on Stefan Brands' proposal [Bra00] and Idemix on the works of Jan Camenisch and Anna Lysyanskaya [CL01, **?**]. Although there are similarities between the two technologies in terms of their constructions and some of their security and privacy properties, there are important differences. They operate with different cryptographic assumptions in different types of groups and they provide different functionality. Most importantly, Idemix provides intrinsically more privacy for the users.

So far, we did not specify what the user's device is. Such a device has to perform the following tasks. First, it has to store secret keys, attributes and credentials. Second, it has to protect this information. Third, it needs to perform all necessary computations securely. Fourth, it has to communicate with the issuers and verifiers. And finally, it has to be usable, that is, 'reasonably easy' to operate with for users.

To select a suitable device in a specific ABC system, one has to carry out a thorough analysis with respect to usage and the possible attack scenarios. Nevertheless, we decide to focus on smart cards because it is a natural choice in terms of security. Typically, a smart card has tamper-resistant secure storage and it is also resistant (to a large extent) against side-channel attacks during cryptographic computations. Furthermore, people are familiar with smart cards and with their various applications (*e.g.* bank cards, public transportation cards, loyalty cards). Finally, the smart-card technology is a viable choice for the user's device, since prototypes have shown [MV11, VA13] that all client-side ABC computations can be implemented on a smart card. The running time of a selective disclosure protocol is around *one second* with the current technology, making ABC-based authentication in many contexts practical.

Besides the benefits of a smart card, there are also inherent challenges. A smart card cannot act on its own and it does not have a user interface. Therefore, it requires an additional device for communication purposes. This is either the user's own device that she has to carry around or it is a potentially untrusted card reader terminal. Such a terminal may maliciously display modified information to the user that she cannot verify. In spite of these difficulties we believe that smart cards are suitable ABC client devices. We address some of the challenges in the rest of the thesis; see terminal certificates in Chapter 4 and Section 6.3, and mobile phones as personal card readers in Section 6.6.2.

The rest of the chapter is organised as follows. First preliminaries and building blocks are described in Section 3.2. Then issuing and verification protocols are explained in Section 3.3. Finally, efficient smart-card implementations are presented in Section 3.4.

3.2 Preliminaries

Before we discuss credential protocols in the next section, we describe the essential building blocks that we will need.[47] The main cryptographic problems and assumptions are discussed in Section 3.2.1, then the original and generalised Pedersen commitment are studied in Section 3.2.2. The last three sections are devoted to the main cryptographic tools to create and use Attribute-Based Credentials: specific credential signatures (Section 3.2.3), zero-knowledge proofs of knowledge (Section 3.2.4) and blind signature protocols (Section 3.2.5).

3.2.1 Cryptographic problems

In this subsection we study groups in which the discrete logarithm (DL) and/or the RSA problem is known to be difficult, and we also define the representation problem.

Discrete logarithm in a prime group The difficulty of solving the discrete logarithm (DL) problem is one of the most fundamental assumptions on which contemporary public-key cryptography can develop primitives. We define it here al-

[47] This section provides a functional description of ABCs without security proofs.

though just in an informal fashion[48]. Let $\mathbb{G}$ be a group in which the group operation can be performed efficiently. Given an element $g \in \mathbb{G}$ that generates cyclic subgroup $\langle g \rangle$ in $\mathbb{G}$ of order q.[49]

Definition 3.2. The ***discrete logarithm problem*** (DL problem) takes the following form: Given $g \in \mathbb{G}$ and $h := g^x$, find $x \in \mathbb{Z}_q$.

The ***DL assumption*** in a given group is the assumption that the DL problem is hard in this group. We will refer to groups as ***DL groups*** in which the DL problem is considered to be computationally infeasible.

Examples of prime groups in which the DL assumption holds include: 1. Let $\mathbb{G} = \mathbb{Z}_p^*$ where p is prime and $g \in \mathbb{G}$ for which the order of $\langle g \rangle$ is $q|p-1$ also prime (their bit-lengths are *e.g.* $|p| = 1024, |q| = 160$); and 2. A group of points with a special point addition operation on an elliptic curve defined over a finite field (of characteristic 2 or a large prime). In both types the order of the cyclic group $\langle g \rangle$ is assumed to be known. (In this chapter we work in integer prime groups with the multiplicative notation. In Chapter 5 we work with an elliptic-curve group.) The parameters for such systems can be described by the group, the generator, the order of the subgroup: $(\mathbb{G}, g, q)$. In the next paragraph we consider groups in which the DL problem is hard but the order of $\langle g \rangle$ is not prime. Actually, the order is hidden in this case.

Strong RSA The strong RSA problem is related to the well-known RSA problem [RSA78], that is, the problem of finding eth root in a group $\mathbb{Z}_n$ where $n = p \cdot q$ (p, q primes).

Definition 3.3. The ***RSA problem*** takes the following form: Given modulus n, exponent $e \in \mathbb{Z}$ and element $y := x^e \mod n$, find $x \in \mathbb{Z}_n$.

This problem is believed to be hard as long as the prime factors p and q are unknown. In fact, these factors are also called a *trapdoor* (or secret key, depending on the context), because they allow the efficient computation of x in the RSA problem.

Note that the RSA problem is to find the unique solution $x \pmod n$ of the equation $y \equiv x^e \pmod n$. The strong RSA problem, introduced simultaneously by Barić–Pfitzmann [BP97] and Fujisaki–Okamoto [FO97], has many solutions because it does not fix the exponent e. The group in which this problem is believed to be hard is the quadratic residue subgroup[50] QR_n of $\mathbb{Z}_n^*$.

[48] See *e.g.* [MOVR97, KL08] for more details.

[49] Note that because the square-and-multiply algorithm requires only at most $\mathcal{O}(\log q)$ group operations, exponentiation can also be efficiently computed.

[50] A quadratic residue r in $\mathbb{Z}_n^*$ resembles a square number in the set of integers: there exists some $s \in \mathbb{Z}_n^*$ such that $r \equiv s^2 \pmod n$. In this thesis, and in Idemix [IBM12], the integer n is a product of two distinct (secret) safe primes p, q, *i.e.* $p = 2p' + 1, q = 2q' + 1$ where p', q' are also primes. The set QR_n of all quadratic residues in $\mathbb{Z}_n^*$ forms a cyclic subgroup of $\mathbb{Z}_n^*$. The order of $\mathbb{Z}_n^*$ is $|\mathbb{Z}_n^*| = (p-1)(q-1) = 4p'q'$ and the order of QR_n is $|QR_n| = \frac{(p-1)(q-1)}{4} = p'q'$, and these orders are assumed to be secret.

Definition 3.4. The ***strong RSA problem*** takes the following form: Given modulus n and $b \in QR_n$, find integers a and e such that $e \geq 2$ and $b \equiv a^e \pmod{n}$.

The *strong RSA assumption* states that it is computationally infeasible to find a solution of the strong RSA problem, as long as the prime factors of n are unknown. As in the case of the RSA problem, the order of the group is assumed to be not known by either of the system's participants except for the entity who knows the prime factors of n.

Since the quadratic residue group is cyclic, there exists a generator $g \in QR_n$ that spans all $\frac{(p-1)(q-1)}{4}$ elements in QR_n. Like in a conventional RSA system, the entity who knows the prime factors of n can efficiently compute[51] the $b^{1/e} \pmod{n}$ for any given $b \in QR_n$ and $e \in \mathbb{Z}$. More interestingly, according to the assumption, in such a group the discrete logarithm assumption also holds: For given $a, b \in QR_n$ it is hard to find e. (If this did not hold, the strong RSA problem could be solved by fixing a.) Therefore, similar cryptographic schemes can be developed in such a group as in a prime DL group.

Representation problem The discrete logarithm problem can be generalised to the representation problem; instead of only one generator, several generators are given in this case.

Definition 3.5. Given a group $\mathbb{G}$ of order q, generated by g, in which the DL assumption holds. Let a tuple of generators $(g_1, \ldots, g_L) \in \langle g \rangle^L$ be fixed. Then the tuple $(x_1, \ldots, x_L) \in \mathbb{Z}_q^L$ is said to be a ***representation*** of H with respect to $(g_1, \ldots, g_L)$ if

$$H = \prod_{i=1}^{L} g_i^{x_i}.$$

The exponents of a tuple of elements uniquely determine the value H. However, given $H \in \langle g \rangle$ and generators $g_1, \ldots, g_L$, there are lots of representations.[52]

Definition 3.6. The ***representation problem*** takes the following form: Given a generator g of the cyclic group $\mathbb{G}$, $(g_1, \ldots, g_L) \in \langle g \rangle^L$ and $H \in \langle g \rangle$, find a representation $(x'_1, \ldots, x'_L)$ with respect to $(g_1, \ldots, g_L)$ (*i.e.* where $H = \prod_{i=1}^{L} g_i^{x'_i}$).

The *representation assumption* in a group is the assumption that the representation problem is hard in this group. First, we observe that this problem can be stated in a prime DL group as well as in a strong RSA setting. Second, it can be proven that

[51] In an RSA group where the modulus is $n = pq$, computations in the exponent are performed $\pmod{(p-1)(q-1)}$ because of Euler's theorem. We will omit modulo notation in the exponents for readability.

[52] Let the order of group $\langle g \rangle$ be $\omega := |\langle g \rangle|$. Then, assuming that the exponents are reduced modulo ω, the number of distinct representations is ω^{L-1}: $L-1$ exponents, say the first $L-1$, can be chosen freely from $\{0, \ldots, \omega - 1\}$ and the last one is uniquely determined by them. Note that computing this last, unique exponent is a DL problem. In this way one gets all representations and all of them are distinct. Using the notation above, in a prime group $\omega = q$, in a strong RSA system $\omega = |QR_n| = \frac{(p-1)(q-1)}{4}$.

finding a representation is as hard as the DL problem[53]. The main idea is that if there is an oracle that can solve the representation problem for any $H \in \langle g \rangle$ and $(g_1, \ldots, g_L) \in \langle g \rangle^L$, we can build a polynomial-time machine that can efficiently solve the DL problem (g, H) in the following way. After selecting L random exponents $(r_1, \ldots, r_L)$ and computing the generators $(g_1 = g^{r_1}, \ldots, g_L = g^{r_L})$, we call the oracle and receive a tuple $(x_1, \ldots, x_L)$ for which $H = \prod_{i=1}^{L} g_i^{x_i}$. Then the discrete logarithm of H can be computed as $x := \sum_{i=1}^{L} r_i x_i$ since

$$H = \prod_{i=1}^{L} g_i^{x_i} = \prod_{i=1}^{L} (g^{r_i})^{x_i} = g^{\sum_{i=1}^{L} r_i x_i} = g^x.$$

3.2.2 Pedersen commitment

A ***commitment scheme*** is a cryptographic primitive used in several protocols, including zero-knowledge proofs and Attribute-Based Credential systems. It enables a party to commit to a value and show the resulting commitment to another party. A commitment *hide*s the value from the receiving party, while it *bind*s the committer to this value, so that the committer cannot change it. A commitment scheme has two phases: *committing* and *opening*. In the committing phase the committer chooses a value a and commits to it by sending $C(a)$ to the receiver. As mentioned, $C(a)$ hides a from the receiver. Later, in the opening phase the committer reveals the value a and possibly provides some additional information required by the scheme. With this information the receiver can confirm that a was indeed the committed value. The binding property ensures that the committer could not have changed the value of a between the commitment and the opening phases.[54]

With respect to the power required to break a given property, both binding and hiding can be *computational* or *perfect* (the latter is also called *information-theoretic*). A commitment that computationally hides a value would not protect this value against a machine that has unlimited computational power whereas a commitment with perfect hiding property would resist even such a machine. Similarly, a scheme with computational binding could not prevent such a machine to change the committed value. In the best case, a scheme provides either perfect hiding with computational binding, or computational hiding with perfect binding.[55]

We show two fundamental examples of commitment schemes both being based on the discrete logarithm problem.

[53] See a detailed proof in Stefan Brands' thesis [Bra00, pp.58–62].

[54] The basic idea of commitment schemes can be demonstrated by a treasure chest as follows. The committer puts a slip of paper with a secret value in it, then locks it with a key, and gives the locked chest to the recipient. On the one hand, the receiver cannot open the box (hiding), on the other hand, he can be convinced that the secret value cannot be changed (binding). Later the committer can reveal the secret value and give the key to the recipient who can verify the committed value.

[55] Perfect hiding and binding in the same scheme cannot be achieved; see [Dam99].

1. Given a DL group setup $(\mathbb{G}, g, q)$, let $C(a) := g^a$. Because of the DL assumption, this commitment hides the value a computationally. However, $C(a)$ perfectly binds the committer to the value a since there is a unique value a' $(\bmod\ q)$ for which $C(a) = g^{a'}$. To open a commitment, the committer simply provides a. Note that if a is taken from a small set, this commitment is inappropriate because the receiver could easily find out a by brute-forcing all possible values.

2. Given a DL group setup $(\mathbb{G}, g, q)$ and an additional generator $h \in \langle g \rangle$. A ***Pedersen commitment*** [Ped92] is defined as $C(a) := g^r \cdot h^a$, where $r \in_R \mathbb{Z}_q$ is chosen uniformly at random. A Pedersen commitment perfectly hides the value $a \in \mathbb{Z}_q^*$ because for any $a' \in \mathbb{Z}_q$ there exists $r' \in \mathbb{Z}_q^*$ such that $C(a) = g^{a'} \cdot h^{r'}$. However, finding $r' \in \mathbb{Z}_q$ after changing $a \in \mathbb{Z}_q$ such that $h^{r'} = g^{a-a'} \cdot h^r$ is a DL problem and thus only computationally infeasible. Therefore, $C(a)$ computationally binds the committer to the value $a \in \mathbb{Z}_q$. To open a commitment $C(a)$, the committer has to reveal both a and r.

We see that to break the binding of a Pedersen scheme, a malicious committer would have to be able to find a different tuple of exponents (a, r) and (a', r') for the same commitment: $C(a) = g^a \cdot h^r = g^{a'} \cdot h^{r'}$. This problem can also be described as follows: Find another representation for $C(a)$ besides (a, r), say (a', r'), with respect to the tuple (g, h). Because of the representation assumption, this problem is hard.

To motivate the following concept, we give some intuition about our application. Our goal is to apply representations to construct Attribute-Based Credentials. Exponents will be considered as attributes. But because attributes may possibly be taken from a small set (*e.g.* , '`gender`', '`nationality`'), the view of a representation problem in which all exponents are attributes could enable a malicious party to brute-force all possible combinations and to find out personal information. Therefore, the direct application of the representation problem would not necessarily hide the attributes. The generalised version of the Pedersen commitment scheme can prevent this issue. [Bra00, BL12]

Definition 3.7. Given a DL group setup $(\mathbb{G}, g, q)$ and additional generators $(g_1, \ldots, g_L) \in \langle g \rangle^L$. The ***generalised Pedersen commitment*** to L values $(a_1, \ldots, a_L)$ is defined as

$$C(a_1, \ldots, a_L) = g^a \cdot \prod_{i=1}^{L} g_i^{a_i},$$

where $a \in_R \mathbb{Z}_q$ is chosen uniformly at random. The *opening* of a generalised Pedersen commitment $C(a_1, \ldots, a_L)$ is $(a, a_1, \ldots, a_L)$.

By introducing an additional exponent a that is randomly chosen from the whole exponent space, we achieve perfect hiding for the rest of the exponents $(a_1, \ldots, a_L)$. The binding property of the generalised Pedersen commitment relies on the difficulty of the representation problem; thus, it is computational.

The generalised Pedersen commitment relies on the discrete logarithm assumption. Definition 3.7 can also be adapted to the strong RSA setting, but the additional exponent a has to be chosen carefully. Because the order of group $\langle g \rangle$ is unknown (in fact, it is $|QR_n|$ as we have seen in footnote 50), the set that a is taken from a very large interval (*cf.* ℓ_n and ℓ_2 in footnote 64).

We show how generalised Pedersen commitments are used to construct credential signatures in the next subsection.

3.2.3 Credential Signatures

We need a specific signature that enables the construction of ABCs. Most importantly, such a signature should sign a block of messages, *i.e.* attributes, instead of only one message. The idea is that the signer will sign a Pedersen commitment in which the exponents are the messages (attributes).

An Attribute-Based Credential can be realised as a (generalised) Pedersen commitment signed by a credential issuer. Knowing the public key of the issuer, any verifier can check the validity of a credential, and thus, of the attributes. Not only verifiability, but also unforgeability and non-repudiation are true for such a digital credential – just like to a conventional signature. Later we show how ABC protocols can be constructed based on the signature; see Section 3.3.

Let the attributes in a credential be denoted by $a_1, \ldots, a_L$, numbers in a certain interval. The encoding of the strings describing the attribute values and converting them into the interval is out of scope.

Brands' signature Brands [Bra00] proposes a signature that can be used for Attribute-Based Credentials. We briefly describe here the signature scheme and later construct a credential protocol with it (Section 3.3).

The signer needs a system with public and private parameters. The signer has two options with regard to the choice of the group he is using. He can generate his own group in which the DL problem is hard, or he can use an existing group $\mathbb{G}$ (*e.g.* in a larger infrastructure with known group parameters). In both cases the signer's system can be described as a DL group, a generator and the order of the generator in the group: $(\mathbb{G}, g, q)$. The signer's private key is $x, x_1, \ldots, x_L \in \mathbb{Z}_q$ and his public key is $h := g^x$ with a public description of the group and the parameters: $(\mathbb{G}, g, g_1, \ldots, g_L)$ where $g_1 = g^{x_1}, \ldots, g_L = g^{x_L}$. Furthermore, the signer selects a (standard) hash function $\mathcal{H}$.

A signature comprises two components. First, it includes a (generalised Pedersen) commitment to the attributes:

$$h' := C(a_1, \ldots, a_L) = g^a \cdot \prod_{i=1}^{L} g_i^{a_i}.$$

Second, it includes a pair of integers $(c', r') \in \mathbb{Z}_q \times \mathbb{Z}_q$. Having the signer's public key, the system parameters and the signature, a verifier can validate the signature as follows:

$$c' \stackrel{?}{=} \mathcal{H}\left(h' \| g^{r'}(h \cdot h')^{-c'}\right).$$

The signature can be viewed in two ways. It is either a signature on the commitment h' or on the block of messages $a, a_1, \ldots, a_L$. In the latter case, the signer relies on the binding property of the generalised Pedersen commitment that guarantees that the attributes cannot be altered.

We emphasise the distinction between the two options because when this signature is used as a credential signature, the issuer actually signs the attributes, while the verifier checks the signature on the commitment. We note that since only the user knows the value a and the issuer does not, the establishment of the signature is an interactive protocol between the issuer and the user; *cf.* a blind signature in Section 3.2.5.

Camenisch–Lysyanskaya signature Camenisch and Lysyanskaya [**?**] propose another signature scheme to construct Attribute-Based Credential, also based on a Pedersen commitment[56]. The Camenisch–Lysyanskaya (CL) signature works in a group in which the strong RSA assumption holds, *i.e.* in the quadratic residue subgroup QR_n of $\mathbb{Z}_n^*$. Because the secret key is specific to the group, each signer has to generate their own group.

The public key of the issuer, which is the output of the key generation algorithm, is the RSA modulus and some random generators from the quadratic residue group QR_n: $(n, Z, S, R, R_1, \ldots, R_L)$. The private key is the (safe) prime factors p, q of n.[57]

Like in a Brands signature, a Pedersen commitment is applied to construct a container for the attributes:

$$R' := C(a_1, \ldots, a_L) = R^a \cdot \prod_{i=1}^{L} R_i^{a_i} \pmod{n}.$$

A CL signature on these attributes is a value A and two randomly chosen values, a prime e and an integer v, *i.e.* (A, e, v), such that:

$$A \equiv \left(\frac{Z}{S^v R'}\right)^{1/e \pmod{\varphi(n)}} \pmod{n}$$

The verification is a rearranged version of the signature in which the eth exponent rather than the eth 'root' is computed; in this respect it is similar to a traditional RSA signature:

$$Z \stackrel{?}{\equiv} A^e \cdot S^v \cdot R' \pmod{n}.$$

[56] This commitment over a hidden-order group was proven to be secure by Fujisaki, Okamoto and Damgård [FO97, DF02]

[57] Additionally, the signer also generates a proof that the parameters were honestly computed. This proof is attached to the public key; *cf.* [IBM12].

Prover Secret: x	$\mathbb{G}, g, q, h = g^x$	**Verifier**
$w \in_R \mathbb{Z}_q$		
$a := g^w$	$\xrightarrow{\quad a \quad}$	
	$\xleftarrow{\quad c \quad}$	$c \in_R \mathbb{Z}_q$
$r := c \cdot x + w \pmod{q}$	$\xrightarrow{\quad r \quad}$	$a \overset{?}{=} g^r \cdot h^{-c}$

Figure 3.3: The Schnorr identification is a proof of knowledge of x: $\mathrm{PK}\{\chi | h = g^\chi\}$.

Randomisation *Randomisation* of a signature is a computation in which a party (not necessarily the signer) modifies a signature without changing the message that it signs. The resulting randomised signature can be verified against the original public key.[58] Brands' signature, to the best of our knowledge, cannot be randomised. This puts some limitations on its use in ABC schemes; see unlinkability in Section 3.3.5.

Using the fact that in a CL signature A is the result of an algebraic computation rather than a hash value (like in Brands' signature), there is a simple and elegant way to randomise a signature. Randomisation is done by first choosing a random integer r (from a large interval) and then by computing a new signature. The randomised signature is $(\overline{A}, e, \overline{v})$ on the same commitment $R' := R^a \cdot \prod_{i=1}^{L} R_i^{a_i} \pmod{n}$, where $\overline{A} := A \cdot S^{-r} \pmod{n}$ and $\overline{v} := v + er$. Indeed, as it can be verified, this (randomised) signature is also valid on R' (that is, on the attributes):

$$\overline{A}^e S^{\overline{v}} R' \equiv A^e S^{-er} S^v S^{er} R' \equiv A^e S^v R' \equiv Z \pmod{n}.$$

3.2.4 Proof of knowledge

We briefly recall the concept of a zero-knowledge proof of knowledge.[59]

Throughout the thesis we will use the notation introduced by Camenisch and Stadler [CS97] to specify zero-knowledge (ZK) proofs in an abstract way. PK stands for *proof of knowledge* and between curly brackets any Greek letters stand for variables that are known to the prover but not to the verifier, all other variables are known to both participants.[60]

For example, Schnorr's proof of knowledge [Sch91] of a discrete logarithm can be described as $\mathrm{PK}\{\chi | h = g^\chi\}$; see Figure 3.3. Both the prover and the verifier know $\mathbb{G}, g, q, h = g^x$ (common input) but only the prover is privy to $x \in \mathbb{Z}_q$ (private input). The protocol runs as follows. The prover commits to a random value

[58] Randomisation is sometimes called blinding; *e.g.* [Ver01].

[59] For a more complete treatment we refer the reader to Goldwasser *et al.* [GMR89] or the tutorials by Damgård `http://preview.tinyurl.com/DamgaardZK` and by Camenisch `http://preview.tinyurl.com/CamenischDAA`, [last accessed: October 26, 2014].

[60] We strive to use Greek symbols similar to those that they represent; such as, χ for x or α_1 for a_1.

$w \in_R \mathbb{Z}_q$ and sends the commitment a to the verifier. The verifier sends a random challenge $c \in_R \mathbb{Z}_q$ to the prover, who responds with r that she can compute by using the challenge c together with her secret key x and random value w. The verification equation $a \overset{?}{=} g^r \cdot h^{-c}$ (in $\mathbb{G}$) only holds if the prover knows x and she computed correctly.

An (honest verifier) zero-knowledge proof of knowledge has to be complete, sound and zero knowledge. Complete: A prover who knows x can convince the verifier, *i.e.* the verification equation $a \overset{?}{=} g^r \cdot h^{-c}$ holds. Sound: If the prover does not know x, she cannot convince the verifier. Zero knowledge: The verifier does not learn any other information just the fact that the prover knows x.

In practice, zero-knowledge proofs are often implemented using the Fiat–Shamir heuristic [FS87]. In this case the proof is not interactive, the challenge c is not provided by the verifier but computed as a hash value of the commitment a from the first step and possibly some message m. First, this turns a proof of knowledge into a signature scheme; *e.g.* Schnorr signature [Sch91]. In this case $c = \mathcal{H}(a \| m), r$ from the non-interactive proof is a signature on m by the prover whose public key is h. The signature can be verified by the equation $c \overset{?}{=} \mathcal{H}(g^r \cdot h^{-c} \| m)$. Second, when such a proof is applied for transactions, m is typically combined from some fixed system description and a fresh, random nonce provided by the verifier that cannot be predicted by the prover. This guarantees that the same proof cannot be replayed in another system or in another transaction.

An ABC selective disclosure protocol is a proof of knowledge in which the credential's secret key and some attributes remain hidden. In this way the verifier is guaranteed that the user owns the credential (*i.e.* she knows all exponents including the secret key); however, the verifier does not learn any information other than the disclosed attributes (see Section 3.3.3 and 3.3.4).

3.2.5 Blind signature

A ***blind signature***, proposed by David Chaum [Cha83], is an interactive protocol between a receiver and a signer that results in a valid signature at the receiver without the signer learning the content of the message or the resulting signature [Bra94]. To achieve this, the receiver first disguises the message before the signer signed it and then the receiver unblinds it by removing the disguise to obtain the final signature. The resulting signature can be verified in the same way as it had been signed without blinding.

In the context of ABCs, blind signatures are used for issuing credentials; the issuer is the signer and the user is the receiver. The user does not reveal her secret key to the issuer but receives a valid credential signature of the issuer on her commitment containing the secret key and the attribute values. Moreover, the issuer does not get to see the resulting signature; thus, he cannot possibly trace the user. Note that the threat of tracing based on the signature is only present if the signature cannot be randomised before verification. As we saw, this is the case at Brands' signatures but not at the CL signatures. Nevertheless, both signatures are issued by applying a blind signature protocol in which neither the secret key nor the resulting signature is known to the issuer (see the next section).

3.3 ABC protocols

Based on the building blocks we described in the previous section, we discuss two constructions that realise Attribute-Based Credentials. Let us recall our objective. Users in an identity management system (IMS) should be able to authenticate in a secure and privacy-friendly way. But unlike in a conventional IMS, in which the user has to authenticate to an identity provider who provides an (ephemeral) security token to access some service, the user can employ a credential dispensed in advance at an issuer and use that to generate a fresh security token 'on the spot' revealing minimal but sufficient information to the verifier. We collect the main security and privacy features and requirements of an ABC system:

(S1) ***Authenticity***. The content of an ABC signed by the issuer cannot be modified and the verifier can verify the signature using this issuer's public key.

(S2) ***Unforgeability*** prevents a malicious third party to forge a valid ABC.

(S3) ***Non-repudiation*** prevents the issuer to deny that the credential's signature was produced by him.

(S4) ***Non-transferability*** prevents the user to transfer her ABC to another user of the system.

(P1) ***Offline issuer***. The issuer of a credential is not involved in the verification protocol.

(P2) ***Issuer unlinkability*** prevents an issuer to trace his credentials. More precisely, an adversary (*e.g.* a colluding set of issuers and verifiers) cannot decide if an issuing protocol and a verification protocol belong to the same credential.[61]

(P3) ***Multi-show unlinkability***. Verifiers cannot trace the activities of a user. More precisely, seeing two verification protocols, no adversary (*e.g.* a colluding set of verifiers and issuers) can distinguish the cases whether those protocols were performed using the same credential or not.

(P4) ***Selective disclosure***. Any subset of attributes from a credential can be revealed and proven independently.

[61] (P2), (P3): Obviously, on the semantic level of attributes, revealed information can make credential activities linkable; *e.g.* disclosing your name in two transactions.

(P5) ***Minimal information***. During verification protocols no other information is revealed to the verifier beyond the disclosed attributes, the credential names and the corresponding issuers.

There are several methods that can realise ABCs based on the DL assumption [Bra00, BL12], the RSA-assumption [Bra00], the strong RSA assumption [**?**], and elliptic-curve DL assumption with pairing for one attribute [Ver01] and for multiple attributes [CL04].

Abstract design process of ABCs Attribute-Based Credentials can be designed following the cryptographic steps below:

1. Construct a suitable signature scheme:
   (a) Consider a commitment scheme;
   (b) Generalise the commitment scheme to a tuple of values instead of only one value;
   (c) Apply a signature on the commitment.
2. Develop ABC protocols based on the signature:
   (a) Use a blinded version of the signature for issuing;
   (b) Apply a proof of knowledge for selective disclosure.

Obviously, this simplified view does not always result in a good Attribute-Based Credential scheme. However, on an abstract level this is the main idea behind both technologies that we discuss in this thesis [Bra00, **?**] and several more cryptographic schemes proposed in the literature, *e.g.* [CL04, BL12, HM13].

Below we describe the issuing and selective disclosure protocols based on the Brands signature, called U-Prove technology [Bra10], and on the CL signature, called Idemix technology [IBM12]. We discussed in Section 3.2.5 that an issuing protocol is a blind signature by the issuer on the user's attributes and secret key. A selective disclosure is a zero-knowledge proof of knowledge of all hidden, non-disclosed attributes from a credential. The number of attributes in a credential in all descriptions is $L \in \mathbb{Z}^+$. To refer to the indices of the disclosed attributes, we will use the notation of a ***disclosing set*** denoted by $\mathcal{D} \subseteq \{1, \ldots, L\}$. Therefore, attributes in a showing instance can be organised in two sets: Attributes $(a_i)_{i \in \mathcal{D}}$ are sent to the verifier before (or during) a protocol run, while attributes $(a_i)_{i \notin \mathcal{D}}$ are not revealed but proven to be in a credential.

3.3.1 Issuing a U-Prove credential

We first describe U-Prove's interactive issuing protocol [Bra00, Paq11].

During the system setup phase an issuer has to generate a DL group resulting in $\mathbb{G}, g, q$.[62] Given the generator $g \in \mathbb{G}$ and the group order q, the issuer chooses a

[62] $\mathbb{G}$ is either a prime group or an elliptic-curve group. We use here multiplicative notation.

Issuer Secret: $x, x_1, \dots, x_L$	$p, q, h, g, g_1, \dots, g_L$ $(a_1, \dots, a_L)$	**User**
$w \in_R \mathbb{Z}_q^*$		
$t := g^w$ in $\mathbb{G}$	$\xrightarrow{\quad t \quad}$	$a \in_R \mathbb{Z}_q^*, \beta, \gamma \in_R \mathbb{Z}_q^*$
		$h' := g^a \cdot h_U$ in $\mathbb{G}$
		$c' := \mathcal{H}(h' \Vert g^\beta (h \cdot h_U)^\gamma \cdot t)$
	$\xleftarrow{\quad c \quad}$	$c := c' + \gamma \pmod{q}$
$r = c\Big(x + \sum_{i=1}^{L} x_i a_i\Big) + w \pmod{q}$	$\xrightarrow{\quad r \quad}$	$t \overset{?}{=} g^r (h \cdot h_U)^{-c}$ in $\mathbb{G}$
		$r' := r + \beta + c'a \pmod{q}$

Figure 3.4: U-Prove's issuing protocol. The resulting signature is $(h', (c', r'))$. (Shorthand notation: $h_U := \prod_{i=1}^{L} g_i^{a_i}$ in $\mathbb{G}$.)

secret key $x \in_R \mathbb{Z}_q$ (uniformly at random) and computes his public key $h = g^x$. Furthermore, he also selects L secret exponents $x_1, \dots, x_L \in \mathbb{Z}_q$ to compute additional generators: $g_1 = g^{x_1}, \dots, g_L = g^{x_L}$.

The credential issuing protocol is a blind signature on the user's attributes. It hides a freshly generated secret key a and the resulting signature $(h', (c', r'))$ from the issuer; see Figure 3.4. Note that the protocol can also be viewed as the issuer's proof of knowledge of his secret key x corresponding to his public key h: $\mathrm{PK}\{(\chi) \mid h = g^\chi\}$. The protocol runs as follows. The issuer commits to a random value w and sends the commitment to the user. The user, instead of just sending a random challenge, constructs c in a way that binds it to the issuer's commitment t, her freshly generated secret key a, and all the attributes $a_1, \dots, a_L$. Moreover, the resulting challenge c enables the user to derive later a valid signature. After receiving c from the user, the issuer computes the response, a modified version of the response of a Schnorr proof of knowledge (Figure 3.3) that also includes the attributes and all his secret exponents. Finally, the user can verify the proof and construct r'. The resulting signature[63] on h' is (c', r') that can be verified as $c' \overset{?}{=} \mathcal{H}\left(h' \Vert g^{r'} (h \cdot h')^{-c'}\right)$. Finally, the user can store the signature which completes the issuance of the credential.

3.3.2 Issuing an Idemix credential

The Idemix issuing protocol is also a blind signature on the user's attributes, the issuer does not learn the user's secret key and the final signature. Unlike U-Prove where all the participants can possibly work in the same group, Idemix requires distinct groups for all issuers. This is essential because each group has only one

[63] We remark that although we say, following Brands, that the signature is *on* h', the real message to be signed is the block of exponents $(a, a_1, \dots, a_L)$. To regard h' as content enables one to verify the signature without knowing the attributes. This is essential at the selective disclosure protocol.

Issuer Secret: p, q	$n, Z, S, R, R_1, \ldots, R_L$ $(a_1, \ldots, a_L)$	**User** Secret: a
		$v' \in_R I(\ell_1)$
$v'' \in_R I(\ell_2)$	$\xleftarrow{U, PK_1}$	$U := S^{v'} R^a \pmod{n}$
Prime $e \in_R [s_1, s_2]$		
$A := \left(\frac{Z}{US^{v''} \prod_1^L R_i^{a_i}}\right)^{1/e} \pmod{n}$	$\xrightarrow{(A,e,v''),PK_2}$	$v := v' + v''$

Figure 3.5: Idemix's issuing protocol comprises two proofs of knowledge; the resulting signature is (A, e, v) on R'. (For simplicity, we denote large intervals as follows: $I(\ell) := \{-2^\ell + 1, \ldots, 2^\ell - 1\}$)

secret key in an RSA setting. Thus, the issuer runs the key generation algorithm. First, a (strong) RSA modulus $n = pq$ is generated, where p', q', $p = 2p' + 1$ and $q = 2q' + 1$ are all distinct primes. Second, a random element of the quadratic residue group modulo n is selected, *i.e.* $S \in_R QR_n$. Third, further $L + 2$ quadratic residues are generated (by raising S to random exponents): $Z, R, R_1, \ldots, R_L \in_R QR_n$. The key generation algorithm outputs the issuer's public key $(n, S, Z, R, R_1, \ldots, R_L)$ and the issuer's private key p, q.

Figure 3.5 shows[64] the issuing protocol in Idemix. We use the abstract notation for the non-interactive proofs of knowledge PK_1 and PK_2. The user proves by PK_1 that she knows the representation of U with respect to the bases (S, R), a randomiser v' and her secret key a:

$$PK_1 := \mathrm{PK}\left\{(\nu', \alpha) \;\middle|\; U \equiv \pm S^{\nu'} R^{\alpha} \pmod{n}\right\}.$$

In the response phase the issuer proves knowledge of $d := 1/e \pmod{\varphi(n)}$:[65]

$$PK_2 := \mathrm{PK}\left\{(\delta) \;\middle|\; A \equiv \pm \left(\frac{Z}{US^{v''} \prod_1^L R_i^{a_i}}\right)^{\delta} \pmod{n}\right\}.$$

Finally, the user checks that the resulting signature is indeed valid: $Z \overset{?}{\equiv} A^e S^v R^a \cdot \prod_1^L R_i^{a_i} \pmod{n}$. The secret key a from the commitment and the value v from the resulting signature (A, e, v) remain hidden to the issuer at the end of the protocol.

[64] For the exact intervals, we refer the reader to the Idemix specification [IBM12, p.40] (some example parameters are as follows: the bit-length of the modulus is $\ell_n = \lceil \log_2 n \rceil = 2048$ and some technical parameters derived from ℓ_n are $\ell_1 = 2465, \ell_2 = 2723, [s_1, s_2] = [2^{596}, 2^{596} + 2^{119}]$).

[65] Since the user does not know whether A is a quadratic residue or not, the proof only demonstrates that A *or* $-A$ is $\left(\frac{Z}{US^{v''} \prod_1^L R_i^{a_i}}\right)^{\delta} \pmod{n}$.

3.3.3 Selective disclosure of a U-Prove credential

A U-Prove selective disclosure is a proof of knowledge of a subset of the attributes in a credential. Abstractly, the selective disclosure has two steps: First, the signature itself is revealed and verified; second, a subset of the attributes is disclosed and proven to be in the credential. Since the credential's signature is revealed and it cannot be changed by the user, it provides only one-time unlinkability; multiple use of a credential is easily linked.

We briefly recall first the verification of a signature. Knowing the system parameters $\mathbb{G}, g, q$, the issuer's public key h and a signature $(h', (c', r'))$ sent by a user, a verifier can check the validity of the signature by performing the following computation: $c' \stackrel{?}{=} \mathcal{H}\left(h' || g^{r'}(h \cdot h')^{-c'}\right)$.

The key observation to construct a selective disclosure protocol can be demonstrated as follows. The Pedersen commitment h', a component in the credential signature, is bound to the credential's secret key and the attributes.

$$h' = g^a \cdot \prod_{i=1}^{L} g_i^{a_i}.$$

If all the exponents $(a, a_1, \ldots, a_L)$ were revealed, the verifier could just 'open' this commitment, that is, he could verify that these are indeed the committed values. However, this is not the case, the secret key and possibly some of the attributes remain hidden. If no attribute is disclosed, the user has to prove knowledge of all the exponents in h'. If all attributes $(a_1, \ldots, a_L)$ are disclosed, $h_U = \prod_{i=1}^{L} g_i^{a_i}$ can be computed by the verifier. The user has to prove knowledge of the discrete logarithm of $g^a = h'/h_U$. Finally, if some attributes are disclosed and others not, the product h_U is logically divided into two terms:

$$h_U = \prod_{i=1}^{L} g_i^{a_i} = \prod_{i \in \mathcal{D}} g_i^{a_i} \cdot \prod_{i \notin \mathcal{D}} g_i^{a_i}.$$

The first product can be computed by both participants, while the second product can only be computed by the user. She proves knowledge of its representation with respect to $(g_i)_{i \notin \mathcal{D}}$.

A selective disclosure proof is the following zero-knowledge proof of knowledge that hides the secret key and all non-disclosed attributes:

$$\mathrm{PK}\left\{ \left(\alpha, (\alpha_j)_{j \notin \mathcal{D}}\right) \;\middle|\; g^{\alpha} \prod_{i \notin \mathcal{D}} g_i^{\alpha_i} = h' \prod_{i \in \mathcal{D}} g_i^{-a_i} \text{ in } \mathbb{G} \right\}.$$

Figure 3.6 shows the realisation of the interactive proof. The user generates $L - |\mathcal{D}| + 1$ random values and commits to them by sending t to the verifier. After receiving a random challenge c from the verifier, the user computes $L - |\mathcal{D}| + 1$ responses $r, (r_i)_{i \notin \mathcal{D}}$ by using the secret key, the hidden attributes, the random values from the

User Secret: $a, a_1, \dots, a_L$	$p, q, h, g, g_1, \dots, g_L$ $(h', (c', r')), (a_j)_{j\in\mathcal{D}}$	**Verifier**
$w \in_R \mathbb{Z}_q, \forall i \notin \mathcal{D} : w_i \in_R \mathbb{Z}_q$ $t := g^w \prod_{i\notin\mathcal{D}} g_i^{w_i}$ in $\mathbb{G}$	$\xrightarrow{t}$	
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_q$
$r = ca + w \pmod q$ $\forall i \notin \mathcal{D} : r_i = ca_i + w_i \pmod q$	$\xrightarrow{r,(r_i)_{i\notin\mathcal{D}}}$	$t \overset{?}{=} g^r \prod_{i\notin\mathcal{D}} g_i^{r_i} \cdot$ $\left(h' \prod_{i\in\mathcal{D}} g_i^{-a_i}\right)^{-c}$ in $\mathbb{G}$

Figure 3.6: U-Prove's showing protocol in which a subset of attributes is disclosed to the verifier. The index set of the revealed attributes is $\mathcal{D}$.

commitment phase and the challenge. The verifier, upon receiving the response r and $(r_i)_{i\notin\mathcal{D}}$, checks the proof using the verification equation. Note that the proof is a generalisation of the Schnorr identification (Figure 3.3): If all attributes are revealed ($\mathcal{D} = \{1, \dots, L\}$) only a is proven to be known in g^a.

Example 3.8. Let us illustrate this proof with two extreme cases: all attributes are disclosed or none. Firstly, the user discloses all the attributes from a credential; so, $\mathcal{D} = \{1, \dots, L\}$. In this case, she sends the attributes $a_1, \dots, a_L$ to the verifier and they perform the following interactive proof in which only the secret key remains hidden:

$$\mathrm{PK}\left\{(\alpha) \;\middle|\; g^\alpha \equiv h' \prod_{i=1}^{L} g_i^{-a_i} \pmod p\right\}.$$

Knowing the attributes, the verifier can compute $h' \prod_{i=1}^{L} g_i^{-a_i}$. The user proves that she knows the discrete logarithm of $h' \prod_{i=1}^{L} g_i^{-a_i}$ with respect to g, that is, the secret key a. Thus, the representation problem that we rely upon here has only one component and therefore it is a discrete logarithm problem.

Secondly, we consider an empty proof, that is, the case where the user proves only the fact that she owns a credential without revealing any attributes from it. Then $\mathcal{D} = \emptyset$ and the user has to prove that she knows a representation of h' with respect to $g, g_1, \dots, g_L$, that is, the secret key and all the attributes:

$$\mathrm{PK}\left\{(\alpha, \alpha_1, \dots, \alpha_L) \;\middle|\; g^\alpha \prod_{i=1}^{L} g_i^{\alpha_i} \equiv h' \pmod p\right\}. \quad \square$$

3.3.4 Selective disclosure of an Idemix credential

The Idemix verification protocol also comprises two parts. First, the user randomises the credential signature, resulting in $(\overline{A}, e, \overline{v})$, and sends $\overline{A}$ to the verifier. (Assume now that the user does not disclose any attribute.) Second, she proves knowledge of all non-disclosed attributes, that is, the representation of the issuer's public parameter Z with respect to $(\overline{A}, S, R, R_1, \ldots, R_L)$:

$$\mathrm{PK}\Big\{(\varepsilon, \overline{\nu}, \alpha, \alpha_1, \ldots, \alpha_L)\Big| Z \equiv \overline{A}^{\varepsilon} S^{\overline{\nu}} \underbrace{R^{\alpha} \prod_{i=1}^{L} R_i^{\alpha_i}}_{R'} \pmod{n}\Big\}.$$

The main idea of the selective disclosure proof is the same as that of U-Prove. In the proof above the disclosing set was empty and the user proved her knowledge of all exponents. If some attributes are disclosed, the verifier can reconstruct some part of the product R', namely $\prod_{i \in \mathcal{D}} R_i^{a_i}$. In this case the user has to provide the following proof:

$$\mathrm{PK}\left\{(\varepsilon, \overline{\nu}, \alpha, (\alpha_i)_{i \notin \mathcal{D}}) \;\middle|\; Z \cdot \prod_{i \in \mathcal{D}} g_i^{-a_i} \equiv \overline{A}^{\varepsilon} S^{\overline{\nu}} R^{\alpha} \prod_{i \notin \mathcal{D}} R_i^{\alpha_i} \pmod{n}\right\}.$$

Note that in the verification there is no distinct signature verification. The fact that the user knows the representation of public key Z (with respect to the appropriate generators) provides sufficient guarantee that she indeed owns a credential signature.

3.3.5 Unlinkability

We briefly analyse unlinkability of user transactions. In this section we assume that disclosed attributes do not make users traceable.

Both selective disclosure protocols provide issuer unlinkability for two reasons. First, the issuer is not included in the verification protocol, so he does not need to learn the fact that his credential is being used. Second, the issuing protocol is a blind signature in which the resulting signature is not known to the issuer. Therefore, even if the verifier and the issuer collude, they cannot correlate issuing and verification instances.

The verification protocols in the two technologies are essentially different in terms of the applied signature. The Idemix technology provides multi-show unlinkability: Because a CL signature can be randomised, a user gives no 'hint' to verifiers that would enable them to trace her. As we saw, the Brands signature, on the other hand, can only be verified if its signature $(h', (c', r'))$ is revealed. Multiple use of the same credential can easily be linked. A pragmatic technique to achieve multi-show unlinkability, and in fact this is proposed in the U-Prove technology overview [Bra10], is to extend the issuing procedure: The user receives not only one but a batch of credentials on the same set of attributes. Having these credentials and making sure that she never shows the same credential multiple times, the user will be truly untraceable. Note that this solution requires the client device to have additional storage space for the extra credentials.

3.4 Performance of smart-card implementations

Smart cards are reasonable candidates to be ABC carriers being tamper resistant, easy to carry around and able to execute cryptographic computations. Indeed, several attribute-based credential schemes and related technologies have been recently implemented on various smart-card platforms [Bic07, Bal08, SGPV09, TJ09, BCGS09, MV11, VA13, BKPR14].

An important application in which the card performs all computations of ABCs autonomously, enables the user perform transactions solely with her card. Our work is mostly based on the implementations described in [MV11, VA13]; see more technical details about the implementations in Vullers' thesis [Vul14]. The card communicates through its contactless interface with the reader and the running times include all communication overhead. The main performance results of the issuing and selective disclosure with credentials of *five* attributes of U-Prove and Idemix are summarised:

- *U-Prove.* The implementation has been done on a MULTOS smart-card platform running on an Infineon SLE66 chip. The operations are performed in a prime subgroup of the prime group $\mathbb{Z}_p^*$ in which the elements are $|p| = 1024$ bits long.

  - *Issuing* The issuing protocol of a credential, containing five attributes (255 bytes each), takes about *5.5 seconds.*

  - *Selective disclosure* A selective disclosure that hides all attributes (so-called empty proof) takes *0.9 second,* while one that hides only two attributes (*i.e.* three attributes are disclosed) runs in *0.6 second.*

- *Idemix.* Idemix has also been implemented on a MULTOS smart card but running on a more recent Infineon SLE78 chip. The operations are performed in an RSA group where the modulus size is $|n| = 1024$.

  - *Issuing* The issuing of a credential, containing five attributes (32 bytes each), takes about *2.6 seconds.*

  - *Selective disclosure* An empty proof runs in *1.5 seconds,* and a proof hiding two attributes from the five in the credential takes just a bit less than *1.0 second.*

Two important conclusions can be drawn from the results above. Attribute-Based Credentials are becoming practical but not yet for all kinds of applications. On the one hand, in identity management systems, such as an eID system [BKPR14] or the ABC ecosystem described in Chapter 6, privacy-friendly authentication is becoming efficient enough relying on personal smart cards carrying full-fledged ABC implementations. On the other hand, more dynamic applications still require faster running times. For instance, in public transportation 0.3 second is considered to be the maximum verification time, whereas this is between 0.6 and 1.5 seconds in these results. Applications that would include both verification and issuing within the same instance take at least 3.5 seconds. This is clearly too much for payments, where an attribute would describe the amount in the user's possession which has to be re-issued with a different amount after a transaction. In sum, currently, attributes are appropriate for rather static scenarios and not for dynamic ones; however, the results are promising and the smart-card technology is probably growing sufficiently fast soon for any applications.

Chapter 4

A Secure Channel for ABCs

> "[W]e assume that the users and organizations are connected by perfectly anonymous channels."
>
> Jan Camenisch and
> Anna Lysyanskaya, 2001

Camenisch and Lysyanskaya make the following assumptions in their important paper [CL01] about anonymous credentials: "Throughout we assume that the users and organizations are connected by perfectly anonymous channels. Furthermore, we assume that for each protocol an organization authenticates itself to the user and that they establish a secure channel between them for each session." The question that we ask in this chapter is how this secure channel can be established between a verifier (an organisation) and an (anonymous) user. Such a channel is crucial to achieve confidentiality for any revealed information and authenticity to protect against adversarial verifier or user behaviour.

4.1 Introduction

In the previous chapter we gave a high-level view of Attribute-Based Credentials (ABCs) and argued that they can be used in practice when the user's side of the protocols are implemented on personal smart cards. Our objective in this chapter is to design a protocol that enables a verifier and a smart card to establish a secure channel. To ensure that appropriate parties communicate with each other, such a channel is set up after mutual authentication. The main challenge is that this protocol should not impair the privacy features of ABCs, such as issuer and multi-show unlinkability (see page 52). Cards have to remain anonymous yet they have to be authenticated.

We focus here on local communication between a smart card and a verifier's terminal. The resulting techniques however can be applied in general for other credential carrier devices and on the top of other channels independently.[66]

[66] In the case when an anonymous secure channel proposed in this chapter is combined with other technologies, the risk of identifiability and profiling of the user on all other layers needs to be analysed separately.

A secure channel is essential for ABC proofs since a selective disclosure (SD) protocol requires protection for the card-holder's privacy against an eavesdropping adversary for various reasons.

- Disclosed attributes: Attribute values are the actual personal information that primarily has to be protected; *e.g.* identifying attributes should not be visible to an eavesdropper.

- Verifier's request: Attribute requests can be considered metadata yet they may reveal valuable and privacy-invasive information; *e.g.* if a video service provider asks for the attribute '`over 18`', it leaks information about the sort of movie a user is watching.

- Issuer's signature: Information about the credential issuer is also metadata that may give hints about the type of credentials and/or the values of attributes; *e.g.* the signature in a credential issued by an employer potentially reveals where an individual works.

Not only eavesdropping but also verifier requests may present threats. A verifier should receive only such personal information that he is entitled to for a particular service or resource. To protect against abusive requests, a verifier should access only those attributes from a card that are necessary for a particular authorisation. In practice, public-key certificates are issued to verifiers that contain a public key and also authorises verifiers to request certain attributes. A card has to verify such certificates, possibly use the verifier's public key for the communication and adhere only to legitimate attribute requests in the selective disclosure proofs.

Active attacks on the user's side can put system security at risk. First, without a secure channel, a classical man-in-the-middle attack can be set up. For instance, acting as a card and using proofs from a real card, a rogue verifier could access some service at another verifier. The same adversary could also inject proofs from other devices that affect the authorisation decision of the verifier. Second, when attributes from multiple credentials are required for an authorisation, the proofs have to be linked to each other. However, ABC's zero-knowledge techniques that securely bind proofs involving multiple credentials are too expensive for some implementation platforms, such as the current smart-card technology. In this case only a secure channel can prevent an adversary to combine proofs originating from different devices. (For instance, a membership attribute on one card could be combined with an '`over 18`' attribute on another card.) A secure channel that is bound to one device on the prover's side can prevent this, so-called card-pooling attack.

A secure channel provides additionally the benefit of a session that can link verification of attributes from already existing credentials on a card and the issuance of new ones. A new credential relies then only on the existence of other credentials of the owner. As a result, an individual can collect new credentials after authentication but potentially without identification.

A tamper-resistant, personal smart card is a viable choice to carry ABCs (*e.g.* , [Bal08, BCGS09, MV11, VA13]). As we saw in the previous chapter, the most essential functionalities are feasible to implement on currently available smart cards. Nevertheless, not all features can be achieved on these resource-limited devices. Due to their complexity, proofs of equality of attributes in separate credentials or property proofs about attributes (like, an attribute lies in an interval or is an element of a set) require more RAM than available on some – otherwise suitable – platforms. This work has mainly been motivated by the Idemix implementation on a MULTOS smart card [VA13] within the I Reveal My Attributes (IRMA) project (discussed in depth in Chapter 6). In this system users' smart cards cannot run equality or property proofs.

Conventionally, to set up a key for a secure channel both participants identify and authenticate each other. However, in the case of ABCs this is not possible. To establish a channel with a verifier, a card cannot reveal a unique, card-specific identifier because it would destroy the privacy properties of the ABC technology. Hence, a new notion of validity is required to realise authentication. A card is regarded as ***valid*** (or authentic) if it holds a particular credential and thus it can perform a proof about it. The choice of this specific credential depends on the system. For example, a national identity card would be considered as such if it can prove that it carries an attribute-based credential issued by the state authority responsible for electronic identity cards. (Note that Direct Anonymous Attestation applies a similar notion of authenticity. [BCC04])

4.1.1 Our Contributions

We introduce a security model for establishing a secure channel between an anonymous ABC card and a terminal (Section 4.3), and we propose two protocols (Sections 4.4 and 4.5) that realise items 1a, 1b, and optionally 3b in the following list of generic steps in the context of ABCs:

1. Perform authentication
   (a) *Establishing a confidential (and semi-authentic) secure channel*
   (b) *Selective disclosure (and authentication) within the channel*
2. Make authorisation decision based on selectively disclosed information
3. Provide service
   (a) Accessing resource, application; or
   (b) [Optional] *Credential issuance within the channel*

We propose to adapt the security model of [BR94] to ABC systems in which the prover's resources are very limited and the user is identified only to the extent of the attribute proofs included. Both protocols we designed are practical and efficient in the sense that the computation and working memory overhead are comparable to the resources ABC protocols require.

4.2 Definition

In the previous chapter we informally described the building blocks of ABCs. Here we define more precisely an ABC scheme for the purpose of the establishment of an underlying secure channel.

Within this model there are three types of participants in an ABC system: an issuer, verifiers and users. Furthermore, we assume that issuers and verifiers receive their public-key certificates, containing their access rights to user attributes, for each application.

Definition 4.1. An ***ABC scheme*** consists of seven probabilistic, polynomial-time algorithms ($\mathsf{Gen}^{\mathsf{Sys}}, \mathsf{Gen}^{\mathsf{I}}, \mathsf{Gen}^{\mathsf{V}}, \mathsf{Issue}^{\mathsf{I}}, \mathsf{Issue}^{\mathsf{U}}, \mathsf{Verify}^{\mathsf{V}}, \mathsf{Verify}^{\mathsf{U}}$) such that:

- $\mathsf{Gen}^{\mathsf{Sys}}(1^k) \mapsto$ System. Taking the security parameter 1^k as input value, the scheme manager runs this algorithm to set up the necessary PKIs, processes and credentials. We assume hereafter that all participants know all public information required to execute their algorithms.

- $\mathsf{Gen}^{\mathsf{I}}(1^k) \mapsto (pk_I, sk_I)$. The issuer runs this algorithm to generate his public key pk_I and private key sk_I.

- $\mathsf{Gen}^{\mathsf{V}}(k) \mapsto (pk_V, sk_V)$. A verifier runs this algorithm to generate his public key pk_V and private key sk_V. The public key is certified by the scheme manager and the certificate also includes a description of the attributes that the verifier is eligible to request from users.

- $\mathsf{Issue}^{\mathsf{I}}(sk_I, pk_I, \text{Attributes}) \mapsto ()$. An issuing instance aims to provide a new credential for a user and it contains two interactive protocols: $\mathsf{Issue}^{\mathsf{I}}$ and $\mathsf{Issue}^{\mathsf{U}}$. $\mathsf{Issue}^{\mathsf{I}}$ is run by the issuer after appropriate verification of the attributes to be issued within the credential in relation to the user. The algorithm takes as input the issuer's key pair and the attributes. For our purposes it suffices to assume that this algorithm does not have output. In an implementation, however, the issuer may want to keep logs of all issuing executions.

- $\mathsf{Issue}^{\mathsf{U}}(pk_I, sk_U, \text{Attributes}) \mapsto$ Cred. The user runs $\mathsf{Issue}^{\mathsf{U}}$, the counterpart interactive protocol of $\mathsf{Issue}^{\mathsf{I}}$. As a result a new ABC is issued on the user's card with the Attributes $a_1, \ldots, a_L$ in it. (Note that Cred is a private output to the user, the issuer does not learn it despite the fact that it contains his signature.)

- $\mathsf{Verify}^{\mathsf{V}}(pk_I, sk_V, \mathcal{D}) \mapsto (b, (a_i)_{i \in \mathcal{D}})$. The verification of an ABC comprises two interactive protocols: $\mathsf{Verify}^{\mathsf{V}}$ and $\mathsf{Verify}^{\mathsf{U}}$. The verifier runs $\mathsf{Verify}^{\mathsf{V}}$ with the issuer's public key, his own secret key and a description of the index set of the required attributes $\mathcal{D}$. At the end of the protocol, $\mathsf{Verify}^{\mathsf{V}}$ outputs bit b (0 – `Reject` or 1 – `Accept`), and the set of requested attributes.

- $\mathsf{Verify}^{\mathsf{U}}(pk_I, pk_V, sk_U, \text{Cred}, \text{Attributes}) \mapsto ()$. This interactive protocol takes all necessary input for an ABC selective disclosure (SD) proof and the verifier's public key (implicitly including the certificate with the description what attributes the verifier is eligible to request).

There are several practical considerations that we briefly discuss here. A CA is a distinct party that does not partake in protocols. There can even be multiple CAs in a scheme and an underlying PKI based on which all participants can verify public-key certificates. For simplicity, we assume that there is only one issuer and all credential types are issued by this party. This makes the description cleaner because we do not need then to manage public (credential) keys of issuers. In practice, there are many issuers and their public keys are certified by CAs, so verifiers and users can check credentials. It does not affect the scheme when it is generalised to multiple issuers. Each verifier's public key pk_V is certified by the CA. The content of a verifier's certificate is similar to a typical X.509 certificate's content (including the name of the verifier, a serial number, the public key, the CA and its signature, *etc.*) with the above-mentioned extension that describes the verifier's attribute access rights. Also, attributes in a selective disclosure proof are always understood as parts of some credentials and thus implicitly describe the credential type and the public key of the corresponding issuer (*cf.* page 41). We do not consider card issuance assuming that each user has exactly one card with a root credential; see more details in Section 6.4.1. In a practical system card provisioning has to be a well-defined process. A credential Cred also includes the secret key sk_U and an expiry date but that is irrelevant in the cryptographic description here. For our purposes it suffices to assume that the $\mathsf{Verify}^{\mathsf{U}}$ algorithm does not have output. In an implementation, however, the user's card should keep a secure log of all verifications.

A verification protocol is generally called a ***selective disclosure*** (SD), as only partial information is revealed from an ABC. As we saw in Sections 3.3.3 and 3.3.4, the protocol is a non-interactive zero-knowledge (NIZK) proof (or so-called signature of knowledge [FS87, CS97]) that signs a message, usually containing a fresh random nonce provided by the verifier. We assume that the ABC technology's proof is correct, sound and zero knowledge. Let us introduce the following, simple notation in which all details (credential type, the issuer's public key, *etc.*) are implicit:

$$\mathsf{SD}\big((a_i)_{i \in \mathcal{D}}; n\big),$$

where $\mathcal{D}$ is the set of indices i corresponding to the disclosed attributes in a given credential and n is the message to be signed by the signature of knowledge. A selective disclosure is called an ***empty proof*** if no attribute is revealed (*i.e.* $\mathcal{D} = \emptyset$). In this case, only the mere existence of a credential is proven.

In order to set up a secure channel for selective disclosure, a card and a verifier have to mutually authenticate each other. However, since being 'valid' is different on both sides, anonymous card authentication and verifier authentication have to be defined separately. A card is considered to be valid (or authentic) if it can show to carry a root credential and perform all eligibly requested selective disclosure proofs. A verifier is valid on the other hand if it owns a valid public-key certificate.

4.3 Secure ABC Channels

To establish a secure channel underlying the ABC proofs, we make use of a former technique. Although many authentication and key-exchange protocols have been proposed at a very early stage of cryptography, Bellare and Rogaway [BR94] are the first who studied authenticated key exchange rigorously. In their model participants are de-coupled and all communication is controlled by an active adversary. To show security of an authentication protocol, one has to prove that the probability for the adversary to make participants accept the other's authenticity is negligible unless all messages are conveyed according to the protocol. The main tool to capture this notion is the so-called *matching conversation*. By attaching some extra information to the mutual authentication protocol, the authors achieve efficient and provably secure key-exchange protocols. The adversary is so powerful that she can query all secret session information from any participants. The security of a key exchange protocol is defined as the indistinguishability of a fresh (not queried) session key from a random string. In spite of the fact that the model has been proposed for a symmetric-key setup, it can be adapted to our setting.

Verifiers in this study are trusted (so they are assumed to be not corrupted), the adversary has access to many cards so that she can query them and even corrupt them. Moreover, all the channels are controlled by the adversary when verifiers and cards are communicating. The security requirements are as follows:

- Selectively disclosed attributes should remain hidden from an eavesdropping adversary;
- Cards should answer queries only to authenticated verifiers;
- Cards should reveal attributes only when a verifier is eligible to request those;
- A verifier should only accept authentication originating from one valid card.

In this section we formalise these objectives.

4.3.1 Security Model

In this model, sessions of cards and verifiers are modelled as oracles that follow the protocol. The adversary is a polynomial-time algorithm that controls the whole communication among oracles. The adversary's goal is to win one of two games, *i.e.* to break one of the following two security properties of the system. First, it tries to eavesdrop on revealed but encrypted attributes thus compromising confidentiality. This is captured by the notion of ***indistinguishability*** of a session key and a random string. Second, the adversary tries to forge authentication by convincing a verifier that he is 'talking' to a valid card while this is not the case. This is captured by the unfeasibility of counterfeiting ***matching conversation***s. If an adversary has only negligible advantage in both games, the protocol is a secure ABC authentication.

We define the system and the challenge for the adversary below.

System initialisation. Given a security parameter k, $1^{\mathsf{Gen}^{\mathsf{Sys}}}$ is run. There are m verifiers and n cards where m and n are polynomials in k. Verifiers run their setup $\mathsf{Gen}^{\mathsf{V}}$ to get their long-term private and public keys, and they receive their certificates $((sk_1, pk_1, cert_1), \ldots, (sk_m, pk_m, cert_m))$. Cards receive their credentials with private keys and sets of attributes $((\alpha_1, C_1, A_1), \ldots, (\alpha_n, C_n, A_n))$ using the issuing protocols $\mathsf{Issue}^{\mathsf{I}}$ and $\mathsf{Issue}^{\mathsf{U}}$. Each card C_i has one secret key α_i which is present in all credentials on this card binding the whole set of attributes A_i to the card. All parties receive their random tapes. Public files describing the whole communication transcript and attributes from corrupted cards are initiated by emptying them: $\mathit{Tr} := \emptyset$, $A_{\mathit{corr}} := \emptyset$.

Cards. A card is considered to be valid if it can perform $\mathsf{Verify}^{\mathsf{U}}$ (including a proof of the root credential) and it has not been compromised yet. The set of all attributes on a card C_i is denoted by A_i.

Verifiers. A verifier is valid if it has a valid public-key certificate $cert_i$ and it can prove knowledge of the corresponding private key. A verifier is only allowed to request attributes in a protocol instance $\mathsf{Verify}^{\mathsf{V}}$ that he is eligible to according to its access policy in $cert_i$. Cards verify the certificate and the access policy, and abort if the verifier tries to query attributes it is not allowed to.

Oracles. Sessions of cards and verifiers are modelled as pairs of oracles. $\Pi^s_{a,b}$ is an oracle modelling party a being engaged in communication with party b in session s.

Adversary. An adversary $\mathcal{A}$ is a probabilistic polynomial-time algorithm that *controls the whole communication* among all verifier and card oracles. It can read, relay, modify, delay or drop messages, inject new ones, or even initiate whole new sessions. It takes 1^k as an input (security) parameter, chooses m, n (polynomials in k) and runs the system initialisation. Then the adversary interacts with the oracles.

Transcript. The whole communication between the adversary and the oracles is described by a public transcript file *Tr* that keeps track of all queries that the adversary asks and all responses that oracles have sent. Interaction between the adversary and an oracle $\Pi^s_{a,b}$ is of the form (τ_i, m_i, r_i) where τ_i is an abstract time moment of the interaction based on which the records are sorted in the transcript, m_i is the query that the adversary submits, and r_i is the response that the oracle returns. After the adversary receives the response, *Tr* is extended: $Tr := Tr \| (\Pi^s_{a,b}, \tau_i, m_i, r_i)$, where $\|$ is the notation for concatenation.

Protocol. A protocol aims to (1) establish a secure channel by setting up a session key $\mathcal{K}$ (of length polynomial in k) to provide confidentiality for attribute requests and selectively disclosed attributes; (2) bind the key exchange and the authentication; and (3) authenticate the two parties to ensure validity. A protocol describes how oracles have to act in a given situation. More precisely, each step depends on the input from the adversary, the oracle's state (that can be deduced from the records of the transcript in which this oracle takes part), its secret session values, and its random tape. An oracle has two special output strings that can be appended to its response r_i at any point of the protocol execution: `KeyOK` and `Accept`. `KeyOK` shows that the oracle established a key as its private output and it is also convinced that its counterpart has established the same key. `Accept` shows that an oracle is convinced that its counterpart is valid. At the end of a protocol run, besides `Accept`, a verifier oracle also outputs a list of attributes, denoted by A, that the card has disclosed. (That is, $A = (a_i)_{i \in \mathcal{D}}$ in the output of $\mathsf{Verify}^{\mathsf{V}}$.) As we already noted, the ABC technology is assumed to be perfectly secure (in terms of unforgeability, non-transferability, unlinkability, etc.).

Adversary's power. There are two types of adversaries. A ***benign adversary*** relays messages according to the protocol (just like a wire). A ***corrupt adversary*** on the other hand can also `Corrupt` a card oracle that has to reveal all secret information it stores: ephemeral secret values of this session, its master secret key and all stored attributes. A protocol will be defined as secure, if the only chance for an adversary to make a verifier accept a card's authenticity is to act benignly.

Matching conversation. For implementation reasons, a verifier is always the ***initiator*** of a conversation and a card is always a ***responder*** in a protocol. Furthermore, the card sends always the last message. A conversation of $\bar{n}$ rounds between a verifier oracle $\Pi^s_{a,b}$ and the adversary can be described then as $(\tau_0, m_0, r_0), (\tau_2, m_2, r_2), \ldots, (\tau_{2\bar{n}-2}, m_{2\bar{n}-2}, r_{2\bar{n}-2})$. A conversation of n rounds between a card oracle $\Pi^t_{b,a}$ and the adversary can be described as $(\tau_1, m_1, r_1), (\tau_3, m_3, r_3), \ldots, (\tau_{2\bar{n}-1}, m_{2\bar{n}-1}, r_{2\bar{n}-1})$. The two lists of tuples are called ***matching conversation*s** if these descriptions meet the following requirements:

1. $\tau_0 < \tau_1 < \tau_2 < \ldots \tau_{2\bar{n}-1} < \tau_{2\bar{n}}$.
2. $m_1 = r_0, m_2 = r_1, \ldots, m_{2\bar{n}-1} = r_{2\bar{n}-2}$.

(Although the indices suggest that no interaction took place among time moments with other oracles, this might not be the case.) An oracle is considered to be ***uncorrupted*** if it has not received the `Corrupt` query.

Revocation. Attributes are revoked by adding them to A_{corr}. By revoking all its attributes, an entire card can be revoked. A revocation mechanism will be assumed to be ***ideal***, that is, as soon as a card is corrupted, it results in immediate card revocation.[67]

Experiment. After specifying security parameter k, we launch the adversary, and let it initialise the system and communicate with oracles. First, it can run the query phase, in which the adversary is allowed to issue q queries in total, where q is polynomial in k. After the query phase, the adversary can select one of the challenges in the test phase. In this phase, it can play one of the following two games.

Confidentiality Game. In the Confidentiality Game the adversary has to gain information about the session key $\mathcal{K}$ (of length a polynomial in k, typically k itself) during a key exchange protocol. In the query phase the adversary is allowed to run oracles and receive their output values. After this phase some of the oracles may have output `KeyOK`. The adversary can run the test phase of the Confidentiality Game at any time by sending $Test(\Pi^s_{a,b})$ where $\Pi^s_{a,b}$ is an uncorrupted oracle. The challenger picks a random bit $b \in_R \{0,1\}$ and sends either the session key $\mathcal{K}$ or a random string of length k to the adversary depending on whether $b = 0$ or $b = 1$, respectively. If the adversary's guess is denoted by b', the advantage of the adversary in the Confidentiality Game is defined as $\mathbf{Adv}^{\mathcal{A}}_{\mathsf{Conf}} = |\Pr[b = b'] - \frac{1}{2}|$ (over all possible session keys).

Definition 4.2. A protocol is said to be a ***confidential key setup*** if the following are true:

1. Under a benign adversary two participants in a matching conversation output `KeyOK` and they both compute the same session key.
2. Any benign or corrupting polynomial-time adversary $\mathcal{A}$ has negligible advantage $\mathbf{Adv}^{\mathcal{A}}_{\mathsf{Conf}}$ to win the confidentiality game.

Authenticity Game. In the Authenticity Game the adversary has to attempt to authenticate to a verifier in an illegitimate way: without a valid card or with multiple cards. We can say that an authentication requires 1 valid card, so the number of those cards must not be < 1 or > 1.

Definition 4.3. Let ***NoMatch*** denote the event that there exist a, b, s, t such that oracle $\Pi^s_{a,b}$ `Accepted` and there is no oracle $\Pi^t_{b,a}$ that engaged in a matching conversation with $\Pi^s_{a,b}$. Let ***Combine*** denote the event that the output list A of a verifier oracle is coming from at least two cards from which (at least) one is uncorrupted, *i.e.* $A \not\subseteq A_i$ (for either of the cards) and $A \not\subseteq A_{corr}$.

The adversary is given oracle access to all instances and is allowed to issue `Corrupt` queries. The adversary wins the Authenticity Game if

[67] We note that because of the ideal revocation assumption, we do not need a non-corrupt adversary that can actively influence the communication by modifying, inserting and deleting messages, but cannot corrupt cards. As soon as a corrupt adversary corrupts a card, all attributes are assumed to be revoked, so no authentication is possible with that card anymore.

1. It manages to trigger the NoMatch event, or
2. It manages to trigger the Combine event.

The probability that the adversary wins the Authenticity Game is denoted as $\mathbf{Adv}^{\mathcal{A}}_{\mathsf{Auth}}$. Roughly speaking, in the former case the adversary succeeds to forge authentication, while in the latter case she is able to combine selective disclosure proofs from multiple cards.

Definition 4.4. A protocol is said to be a ***secure ABC authentication*** if the following are true:

1. Under a benign adversary a card and a verifier can mutually verify each other's validity in a matching conversation; and
2. Any corrupting polynomial-time adversary has negligible advantage $\mathbf{Adv}^{\mathcal{A}}_{\mathsf{Auth}}$ to win the authenticity game.

Therefore, if a verifier accepts an ABC showing proof, the selectively disclosed attributes are originating from *one valid* card. The same can be expressed by the sets of attributes:

Corollary 4.5. If a secure ABC authentication protocol is given by which a card successfully authenticates to a verifier, then either $A \subseteq A_{corr}$ or $A \subseteq A_i$ (where $A_i \cap A_{corr} = \emptyset$).

4.4 Implicit Card Authentication

We introduce an efficient secure ABC authentication protocol in Figure 4.1, which is called the implicit card authentication protocol or ***ICA***. The verifier and a presumed card establish a key $\mathcal{K}$ that is used to provide a secure channel based on $\mathcal{K}$ for the selective disclosure proofs. Note that unlike most authenticated key exchange protocols, the card's validity can only be verified within the channel, which explains the name of this protocol.

We assume that the verifier's public key is an initial input value to the card. Moreover, the card is also privy to the description of which attributes this verifier is eligible to request. In practice, a public-key certificate is sent to the card, from which it can extract and verify pk_V and the attribute access rights. The verifier's private initial input is its secret key sk_V. We assume that the ciphertext space of the public-key encryption $\mathcal{E}()$ is identical to the key space of the symmetric-key encryption Enc(). Furthermore, we also assume that all participants have access to random oracles f_1, f_2 and f_3.

A verifier initiates the protocol ***ICA*** by sending a random nonce $n_V \in_R \mathcal{N}$ chosen uniformly at random from the nonce space $\mathcal{N}$. Then, the card also generates a random nonce $n_C \in_R \mathcal{N}$ from the same nonce space. It encrypts n_C using the public key pk_V and sends $\mathcal{E}_{pk_V}(n_C)$ to the verifier. After receiving $\mathcal{E}_{pk_V}(n_C)$,

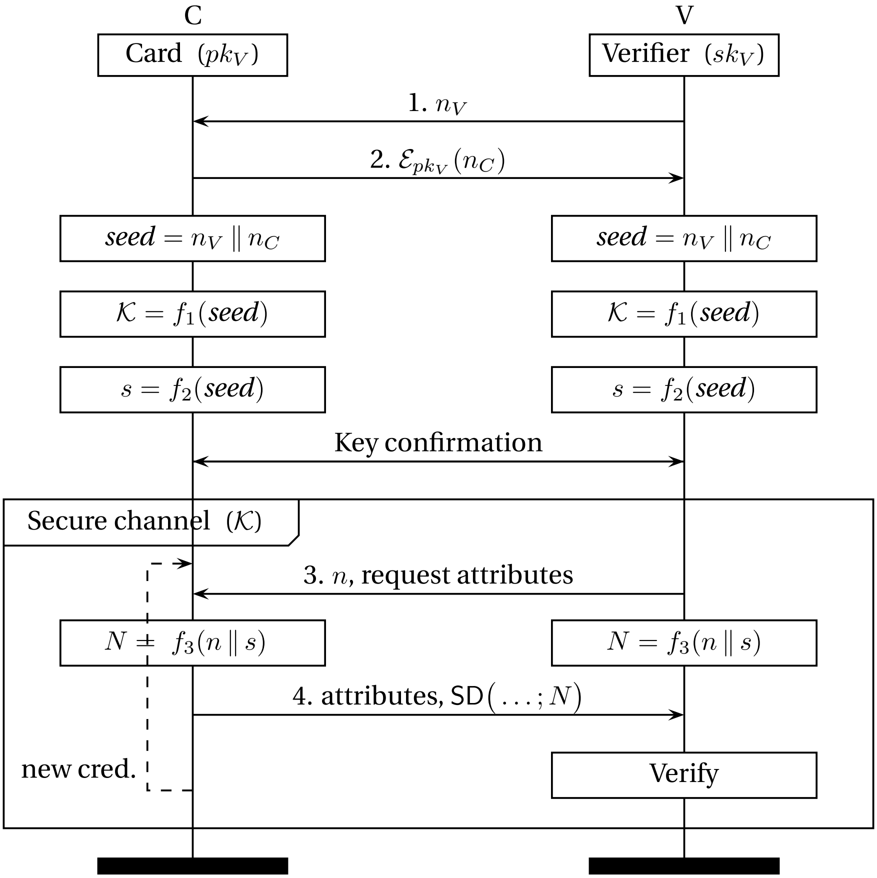

C
V
Card (pk_V)
Verifier (sk_V)
1. n_V
2. $\mathcal{E}_{pk_V}(n_C)$
seed $= n_V \parallel n_C$
seed $= n_V \parallel n_C$
$\mathcal{K} = f_1($*seed*$)$
$\mathcal{K} = f_1($*seed*$)$
$s = f_2($*seed*$)$
$s = f_2($*seed*$)$
Key confirmation
Secure channel $(\mathcal{K})$
3. n, request attributes
$N = f_3(n \parallel s)$
$N = f_3(n \parallel s)$
4. attributes, $\mathsf{SD}(\ldots; N)$
new cred.
Verify

Figure 4.1: Implicit card authentication ***ICA***: A key exchange for ABC selective disclosure precedes card authentication.

the verifier can decrypt it and compute n_C. Both participants can now compute *seed* $= n_V \| n_C$ and store the derived values: the channel key $\mathcal{K} = f_1($*seed*$)$ and the binder $s = f_2($*seed*$)$. Note that only the verifier has already authenticated – being required to use its private key. Explicit key confirmation ensures both participants that they share the same secret key $\mathcal{K}$. In practice, the *seed* has to be deleted on both sides.

Within the established secure channel the two parties perform the selective disclosure proofs that eventually provide card authenticity. To initiate each selective disclosure, the verifier sends a fresh, random nonce $n \in_R \mathcal{N}$ and requests some attributes to be disclosed. A verifier is allowed to request a set of attributes residing in *one* credential since cards are not assumed to be able to link proofs from different credentials. The card checks that the verifier is entitled to request the attributes.

If not, the card rejects the request. Otherwise, it computes $N = f_3(n \| s)$ and using it as a fresh nonce, the card generates a 'normal' ABC non-interactive selective disclosure proof that reveals the requested attributes and proves that they reside in the credential. The verifier checks the proof using the same N and stores the disclosed attributes in A. After all credential proofs with the required attributes were requested and performed successfully, the verifier terminates and outputs `Accept` and the set A of revealed and verified attributes.

4.4.1 Security Analysis

Let us consider the key establishment phase of protocol ***ICA*** and denote it as $\boldsymbol{ICA}^{key}$. It establishes a shared secret key $\mathcal{K}$ between two parties in the presence of a *corrupt adversary* as follows.

$\boldsymbol{ICA}^{key}$ protocol:

1. $C \leftarrow V$: n_V
2. $C \rightarrow V$: $\mathcal{E}_{pk_V}(n_C)$
3. C, V: $\mathcal{K} := f(n_V \| n_C)$
4. C, V: Key confirmation

During the system initialisation both participants receive the security parameter 1^k and their own random tapes. Also, V's key pair (pk_V, sk_V) is generated, and C receives pk_V, V receives (pk_V, sk_V). To start a protocol run, the verifier generates a nonce $n_V \in_R \mathcal{N}$ (uniformly at random) and sends it to C. Then C also generates a random nonce $n_C \in_R \mathcal{N}$, encrypts it with V's public key and sends $\mathcal{E}_{pk_V}(n_C)$ to V. Using sk_V, V can compute n_C. Both parties, having access to a random oracle f, can compute the shared key $\mathcal{K}$ from n_V, n_C by $\mathcal{K} := f(n_V \| n_C)$. Finally, after explicit key confirmation, both parties output `KeyOK`.

Note that this key-setup protocol can not provide card authenticity because there is nothing that would authenticate C. The only thing we wish to prove that the protocol is *confidential*, that is, seeing only the messages, an adversary is not able to gain information about the established key $\mathcal{K}$.

Lemma 4.6. *Assume that $\mathcal{E}()$ is a CPA-secure public-key encryption,* $\mathrm{Enc}_{\mathcal{K}}(\cdot)$ *is a semantically secure (symmetric-key) encryption, and f is a random oracle. Then $\boldsymbol{ICA}^{key}$ is a confidential key setup in the presence of a corrupt adversary.*

Proof. We prove that if we have access to an adversary $\mathcal{A}^{IK}$ that can break the security of $\boldsymbol{ICA}^{key}$, then we can build a polynomial-time algorithm $\mathcal{A}^{pke}$ that can break the semantic security of the public-key encryption $\mathcal{E}()$. (The key generation algorithm of $\mathcal{E}()$ is denoted by $\mathcal{G}_{pke}$. Its output is a public-private key pair.)

First we describe the operations of both adversaries.

- Adversary $\mathcal{A}^{IK}$ can determine the verifier's nonce and learn the card's output (the encrypted nonce). The adversary's task is to distinguish the true established key from a random value (in the key space). The experiment with $\mathcal{A}^{IK}$ goes as follows:

  1. The challenger runs key generation $\mathcal{G}_{pke}$ and sends pk_V to $\mathcal{A}^{IK}$;
  2. $\mathcal{A}^{IK}$ outputs $n_V \in \mathcal{N}$;
  3. The challenger does the following:
     (a) Generates nonce $n_C \in_R \mathcal{N}$ and encrypts it;
     (b) Generates a random bit $\hat{b} \in_R \{0,1\}$, computes the challenge ch:
     $$ch := \begin{cases} f\left(n_V \| n_C\right) & \text{if } \hat{b} = 0 \\ \text{random (of length } k\text{)} & \text{if } \hat{b} = 1; \end{cases}$$
     (c) Sends $\mathcal{E}_{pk_V}(n_C)$ and ch to $\mathcal{A}^{IK}$;
  4. $\mathcal{A}^{IK}$ queries the random oracle f (at most polynomial times in k);
  5. Finally, $\mathcal{A}^{IK}$ sends her guess $\hat{b}'$.

  $\mathcal{A}^{IK}$ should have the power to corrupt cards. This is modelled by allowing $\mathcal{A}^{IK}$ to send a `Corrupt` query at any point in Steps 3 and 4. As a response the challenger reveals n_C and $f\left(n_V \| n_C\right)$. In this case the adversary has to go to Step 2 in the algorithm, to query a 'new card'. (The adversary is allowed to submit at most q `Corrupt` queries where q is polynomial in k.)

- Adversary $\mathcal{A}^{pke}$ tries to attack the ***CPA-security*** of the public-key encryption $\mathcal{E}()$:

  1. The challenger runs $\mathcal{G}_{pke}$, stores the secret key and sends pk_V to $\mathcal{A}^{pke}$;
  2. $\mathcal{A}^{pke}$ sends two messages m_0, m_1 of the same length to the challenger;
  3. The challenger picks a random bit $b \in_R \{0,1\}$, computes the ciphertext $c = \mathcal{E}_{pk_V}(m_b)$ and sends c to $\mathcal{A}^{pke}$;
  4. Finally, $\mathcal{A}^{pke}$ outputs a guess b'.

We show that if the advantage of $\mathcal{A}^{IK}$ is not negligible in the first experiment then the advantage of $\mathcal{A}^{pke}$ is also non-negligible in the second experiment. As usual, we show how adversary $\mathcal{A}^{pke}$ can use $\mathcal{A}^{IK}$ to break the security of the public-key encryption.

1. The challenger runs $\mathcal{G}_{pke}$, stores the secret key and sends pk_V to $\mathcal{A}^{pke}$; $\mathcal{A}^{pke}$ sends pk_V to $\mathcal{A}^{IK}$;

2. $\mathcal{A}^{IK}$ sends n_V to $\mathcal{A}^{pke}$;

3. $\mathcal{A}^{pke}$ generates random nonces $n_C, \widetilde{n}_C \in_R \mathcal{N}$ and sends them to the challenger;

4. The challenger answers with a ciphertext c encrypting one of the nonces;
5. $\mathcal{A}^{pke}$ computes $\mathcal{K} := f\,(n_V \| n_C)$ and sends $(c, \mathcal{K})$ to $\mathcal{A}^{IK}$;
6. $\mathcal{A}^{IK}$ sends random-oracle queries (at most polynomially many in k) to $\mathcal{A}^{pke}$ to which $\mathcal{A}^{pke}$ replies as a random oracle;[68]
7. $\mathcal{A}^{IK}$ replies with a guess $\hat{b}$;
8. $\mathcal{A}^{pke}$ forwards the guess to the challenger, *i.e.* $b' := \hat{b}'$.

If $\mathcal{A}^{IK}$ sends a `Corrupt` query at any Step, $\mathcal{A}^{pke}$ answers with $n_C, f\,(n_V \| n_C)$ and $\mathcal{A}^{IK}$ has to go to Step 2. In this case, $\mathcal{A}^{pke}$ sends a random guess b' to the challenger.

In Step 5 there are two possibilities for the pair $(c, \mathcal{K})$ depending on whether the challenger encrypted n_C or $\widetilde{n}_C$. It is either $(\mathcal{E}_{pk_V}(n_C), f\,(n_V \| n_C))$ or $(\mathcal{E}_{pk_V}(\widetilde{n}_C), f\,(n_V \| n_C))$ – and $\mathcal{A}^{pke}$ does not know which is the case. In the former case the session key corresponds to the encryption, while in the latter case $f\,(n_V \| n_C)$ is just a random value in the light of the two nonces $n_V, \widetilde{n}_C$. Every time when $\mathcal{A}^{IK}$ guesses correctly, $\mathcal{A}^{pke}$ also guesses correctly. If $\mathcal{A}^{IK}$ guesses incorrectly, $\mathcal{A}^{pke}$ has negligible advantage. Finally, when $\mathcal{A}^{IK}$ corrupts, $\mathcal{A}^{pke}$ just randomly guesses, which also means negligible advantage. Therefore, the advantage of $\mathcal{A}^{pke}$ against the public-key encryption scheme is at least $1/q$ times the advantage of $\mathcal{A}^{IK}$ against ***ICA***^{key}.

The algorithm above is a perfect simulation of the key establishment for adversary $\mathcal{A}^{IK}$ and it finishes in polynomial time. Since the $\mathcal{E}()$ is CPA-secure and the advantage of $\mathcal{A}^{pke}$ can only be negligible, the advantage of $\mathcal{A}^{IK}$ is also negligible (though q times as much) and thus ***ICA***^{key} is a confidential key setup. □

Because adversary $\mathcal{A}^{IK}$ cannot distinguish a fresh key from a random string (of the same length), the ciphertext, encrypted with a fresh key, is indistinguishable from the ciphertext encrypted with the random string. Therefore, the communication, encrypted with the semantically secure $\mathrm{Enc}_{\mathcal{K}}()$, after this key establishment is indistinguishable from random communication for the adversary. (We observe that in ***ICA*** s has the same property as the session key $\mathcal{K}$ as it is set up in the same way using an independent random oracle.)

Corollary 4.7. ***ICA*** is a confidential key setup.

Now we have to show that ***ICA*** is authentic, that is, both the card and the verifier get convinced about the other's validity. The first is easier, since the verifier initiates the interrogation by sending its certificate with his public key. The card checks validity of the public key in the certificate. Afterwards, it uses the public key in ***ICA***^{key} to set up the session key $\mathcal{K}$. Using an explicit key confirmation subsequently,

[68] Operations of a random oracle: Start with an empty look-up table and answer queries with values of fixed length as follows. 1. Generate random output to a new input value and store the pair of input and output values in the look-up table; 2. Return the same output value as before if a queried value occurs in the look-up table.

the card can be guaranteed that the verifier is the owner of this valid public key, and as a result, the verifier has authenticated to the card.

The second task is to show that the card also authenticates to the verifier and an adversary's only possibility to successfully authenticate to a verifier is to behave benignly and relay all messages unchanged between a valid card and a verifier. More specifically, to prove authenticity of ***ICA***, we need to show that the probabilities of NoMatch and Combine events are negligible.

The impossibility of NoMatch relies on the security of the ABC system. NoMatch means that the adversary could create a set of valid ABC proofs (one of them based on the root credential) without having a valid card that produced that. But this is impossible because of the unforgeability of credential proofs in the ABC technology.

The Combine event is triggered if selective disclosure proofs from more than one card are combined into one proof. The intuition behind the way we prove the unfeasibility of Combine is as follows. First, we give the adversary power to learn the session key and control all messages as they were sent in the clear. Second, we define the so-called *binding* property that provides two guarantees. Particular SD proofs are bound to each other in one ABC authentication and they are bound to the key-setup phase. This ensures that the device participated in the key establishment is the same as the one that produced all the proofs.

According to the Fiat–Shamir heuristic [FS87], a NIZK proof can be considered as a signature that can sign an arbitrary message. When such proofs are used as an authentication mechanism, this message often contains a nonce chosen randomly by the verifier. The binding property is defined by a game similar to a MAC existential forgery game, where transformed nonces play the role of MAC tags. Having queried messages (nonces) during the query phase to an oracle, it outputs transformed nonces of each of them. The adversary tries to create a new, valid pair of a nonce and its transformed version in the test phase.

Binding Game. Let the nonce space be the same as the key space, *i.e.* $\mathcal{N} = 2^{|\mathcal{K}|}$. The nonce transformation is described by the map $B : \{0,1\}^{|\mathcal{N}|} \times \{0,1\}^{|\mathcal{N}|} \to \{0,1\}^{|\mathcal{N}|}$, where B has two input values, a secret value (*seed*) and a public value (a nonce). The adversary's goal is to produce an existential forgery of a valid output nonce N on any nonce n of her own choice. In this game the adversary receives even the session key $\mathcal{K}$ as input; this models her power to eavesdrop on and inject in message flows in the secure channel. The two phases of the game go as follows. The query phase comprises q rounds (where q is a polynomial in k). In a round the adversary can choose a nonce n of her choice and the challenger answers with $N = B(\mathit{seed}, n)$. In the guess phase the adversary has to output a valid pair of a nonce and a transformed nonce $(\hat{n}, \hat{N})$. The adversary wins the Binding Game if $(\hat{n}, \hat{N})$ is valid and has not occurred in the query phase. The advantage $\mathbf{Adv}^{\mathcal{A}}_{\mathsf{Bind}} = \Pr[\text{Adversary wins}]$.

Note that we require little from the adversary while she is very powerful. She receives the session key $\mathcal{K}$ and she can query many transformed nonces; she receives $N = B(\mathit{seed}, n)$ for each nonce n of her own choice. She is not required to produce a valid selective disclosure proof, only a new valid pair $(\hat{n}, \hat{N})$.

Definition 4.8. A confidential key setup is said to provide ***secure binding*** if any polynomial-time adversary has negligible advantage $\mathbf{Adv}^{\mathcal{A}}_{\mathsf{Bind}}$ in the Binding Game.

In ***ICA*** $N = B(n) := f(n\|\mathit{seed})$ binds the key-setup phase and the nonces. We show that this binding is appropriate in general and does not damage the assurance of freshness. Distributions 2. and 3. in the following lemma model the card's and the adversary's views of $f(n\|\mathit{seed})$.

Lemma 4.9. *Let the size of the nonce space and the output of a random oracle f be $\mathcal{N}$, a polynomial in the security parameter (so, the random oracle $f : \{0,1\}^* \to \mathcal{N}$). Then the distributions of the following random variables are indistinguishable, where n, $\mathtt{str} \in \mathcal{N}$ are uniformly distributed random variables:*

1. *n;*

2. *$f(n\|\mathtt{str})$, where n is given; and*

3. *$f(n\|\mathtt{str})$, where $\mathtt{str}$ is given.*

Proof. Assume that there is a polynomial-time algorithm that can distinguish the distribution of 1. and 2. That would give us a statistical test for $f(\cdot)$, which contradicts to the assumption that it is a random oracle. Similarly, 1. and 3. are indistinguishable.

The distributions of $f(n\|\cdot)$ and $f(\cdot\|\mathtt{str})$ are the same because $|n| = |\mathtt{str}|$ and f is a random oracle. □

Corollary 4.10. The probability of guessing a randomly chosen value correctly in case of any distribution in Lemma 4.9 is negligible in the security parameter k (in fact, the probability is $1/2^{|\mathcal{N}|}$).

Corollary 4.11. $\mathsf{SD}\big(\ldots; f(n\|\mathtt{str})\big)$ provides freshness and binding.

Proof. Freshness: If n is unpredictable for the card, then $f(n\|\mathtt{str})$ is also unpredictable by the equivalence of distributions 1. and 3. in Lemma 4.9.

Binding: The adversary has to produce a new pair of $(n, f(n\|\mathtt{str}))$ for any $n \in \mathcal{N}$ after at most q queries without knowing $\mathtt{str}$. By distribution 2. in Lemma 4.9 and by Corollary 4.10, the probability for the adversary to guess correctly is negligible in the security parameter k ($\mathbf{Adv}^{\mathcal{A}}_{\mathsf{Bind}} \leq \frac{1}{2^{|\mathcal{N}|}-q}$). □

As we will see from the next lemma, it is necessary to double the nonce space to keep the same security level of the system.

Lemma 4.12. *If $|\mathcal{N}| \geq 2k$, then the probability of* $\mathsf{Combine}$ *is negligible.*

Proof. If the adversary wants to trigger Combine, she has two disjoint possibilities: To start with an uncorrupted card to get it produce a proof or to start with a corrupted card to do that.

First case: The first selective disclosure proof $\mathsf{SD}\big(\dots; f(n_1 \| \mathit{seed})\big)$ is provided by an uncorrupted card C. Thus, the adversary cannot know *seed* unless it corrupts this card. By Corollary 4.11, the adversary has negligible probability to guess the transformed nonce correctly and thus to trigger Combine. On the other hand, if the adversary decides to corrupt C, this case in fact becomes the second case.

Second case: We can assume that the adversary knows all secret values of the session: $\mathcal{K}$, *seed*. Furthermore, she already has a few selective disclosure proofs with attributes from A_{corr} (*i.e.* from a corrupted card). In order to trigger the Combine event, she has to use an uncorrupted card C. Therefore, she needs a proof $\mathsf{SD}\big(\dots;$ $f(n'_i \| \mathit{seed}')\big)$ from C where $f(n'_i \| \mathit{seed}') = f(n_i \| \mathit{seed})$, provided that she has a shared secret *seed* with the verifier oracle which sent nonce n_i. Since f is a random oracle, a collision can happen roughly with $2^{-|\mathcal{N}|/2}$ probability (because of the birthday paradox).

In sum, the adversary's advantage is about $2^{-|\mathcal{N}|/2} \leq 2^{-k}$ which is negligible. □

Lemma 4.13. *If* $|\mathcal{N}| \geq 2k$*, then* ***ICA*** *is a secure authentication.*

Proof. We first consider mutual authentication, then we show that the adversary has only a negligible probability to win the Authenticity Game (page 69).

Verifier authentication: According to Lemmas 4.6 and Corollary 4.7, a card can output `KeyOK` and `Accept` at the end of the key-setup phase. Card authentication: A verifier can `Accept` by the end of the selective disclosure protocol that there is a valid card in a matching conversation. It also can output A.

We have to show that no adversary can trigger the NoMatch or Combine events. Assume that an adversary does trigger the NoMatch event, *i.e.* she can authenticate to a verifier, but there is no matching conversation with a card. By Corollary 4.5 the verifier can be convinced that either all attributes are coming from corrupted cards, or all of them are coming from *one valid* card. The verifier performs the following steps. It checks all selective disclosure proofs in terms of the disclosed attributes and their proofs, their freshness, and that they include the card validity credential. Then the verifier checks each attribute whether it is in A_{corr}. If at least one of them is in the set of corrupted card attributes, the verifier can be convinced that the whole proof is made up of attributes from corrupted cards. Otherwise, the attributes originate from one valid card.

The other chance that the adversary has is to trigger the Combine event. As Lemma 4.12 shows, this event has probability about $2^{-|\mathcal{N}|/2} \leq 2^{-k}$ (all other events' probability is of the order $2^{-|\mathcal{N}|}$) and thus, the advantage of the adversary is negligible. □

Now we are ready to prove that ***ICA*** is a secure ABC session protocol.

Theorem 4.14. *Assume that a CPA-secure public-key encryption* $(\mathcal{G}_{pke}, \mathcal{E}(), \mathcal{D}())$ *is given to encrypt confidential messages to verifiers, a secure ABC technology is given for selective attribute disclosure with ideal revocation, and a CPA secure encryption* $\mathrm{Enc}_{\mathcal{K}}(\cdot)$ *is given for secure channels with arbitrary keys from the key space. Moreover, let* f_1, f_2, f_3 *be random oracles with all participants (including the adversary) having access to them and assume that* $|\mathcal{N}| \geq 2k$ *where* k *is the security parameter of the system. Then the implicit card authentication protocol* ***ICA*** *is a secure ABC authentication protocol.*

Proof. Clearly, two parties following the protocol can both `Accept` under a benign adversary. Furthermore, the verifier receives all eligibly requested attributes that it can output at the end of the protocol.

By Corollary 4.7 and by Lemma 4.13 together with the assumption that $|\mathcal{N}| \geq 2k$, $\mathbf{Adv}^{\mathcal{A}}_{\mathsf{Conf}} + \mathbf{Adv}^{\mathcal{A}}_{\mathsf{Auth}}$ is negligible.

Hence, ***ICA*** is a secure ABC authentication protocol. □

4.4.2 Practical Considerations

Having proven that the ***ICA*** protocol is secure (assuming some standard primitives), we briefly discuss practical considerations.

As we already mentioned, the verifier's public key pk_V is not an initial input value, rather it is a part of a certificate in the first flow. The signature on the certificate within one ABC system desirably belongs to one authority, making it more efficient for the card to verify only one signature. An alternative, less efficient solution is to build a PKI of certificate authorities. Furthermore, if we assume that random oracles are instantiated as standard hash functions $\mathcal{H}$(such as, SHA-2 or SHA-3), ***ICA*** entails a few hashes on the card's side. Presuming that an RSA signature scheme is applied for the certificates and RSA encryption is used to encrypt the nonce n_C for verifiers, the key setup of ***ICA*** (*i.e.* ***ICA***^{key}) is also efficient, as it requires only two exponentiations (a signature verification and a public-key encryption) and two hashes (during signature verification and creating a session key). In ***ICA*** each credential proof requires an additional hash over the ABC computations.

Although the security proof does not require the symmetric-key encryption to be authenticated, we propose to use authenticated encryption after the key establishment phase between a card and a verifier. An ABC proof gives certainty about integrity, any changes prevent the proof to be verifiable. However, standard authentication mechanisms guarantee easy handling of errors during flow transmission. Authenticated encryption can be achieved either by using an efficient primitive, such as OCB [RBB03], or by a card- and implementation-specific solution – in which for instance, APDU messaging is already authenticated with a given MAC key and messages are sent encrypted with an encryption key. Depending on the authenticated encryption mechanism, keys and initiating values have to be chosen appropriately. In any case, they should be derived from the *seed* using a hash function. For instance, when separate encryption and authentication keys are required, they can be computed as $\mathcal{K}_{enc} = \mathcal{H}(\texttt{0x00}\|\textit{seed})$ and $\mathcal{K}_{MAC} = \mathcal{H}(\texttt{0x01}\|\textit{seed})$. An explicit key confirmation is recommended after both parties can compute $\mathcal{K}$. However, this step can be merged with the first message within the secure channel in practice if the format of messages is sufficient to prove that the proper key is used. For instance, the attribute request may be required to start with 80 zero bits within the (authenticated) secure channel.

Nonces are not only necessary to prevent replay attacks in ***ICA***, but they bind also selective disclosure proofs to the key establishment, that is, to the secure channel. Therefore, nonces should be long enough to protect also against the birthday attack and thus the nonce space $\mathcal{N}$ should be $\{0,1\}^{128}$ (or larger).

We proposed to use a root credential that is only issued to valid cards after a rigorous verification procedure. The presence of such a credential, which can be demonstrated using an empty proof, shows that a card is valid. However, since ABC proofs are rather expensive in terms of time, it is often desirable to omit as many selective disclosure proofs as possible. In a slightly modified trust model, verifiers may rely on issuers to verify properly the root credential before they issue new Attribute-Based Credentials. In this case verifiers do not need to request a separate validity proof. This is a decision that the given system manager and/or a particular verifier can decide upon. Needless to say, if the verification aims at a service of issuing new credentials on a particular card, card validity has to be verified and issuance has to be carried out in the same secure channel as the verification.

4.5 Diffie–Hellman ABC Channel (ABCDH)

Although ***ICA*** is an efficient protocol to build a secure channel for ABC proofs, in some applications it might be desirable to make an authentic (not only confidential) key establishment. Furthermore, the roles of a verifier and a card is very asymmetrical in ***ICA***, while in future scenarios participants may be provers and verifiers simultaneously. For instance, two, possibly anonymous NFC-enabled phones authenticate each other using some ABC technology. In this case their roles are symmetrical, both participants are provers as well as verifiers. We propose a protocol that addresses these issues.

Unlike in ***ICA***, in this protocol authentication of both parties happens in the key establishment phase. But we also borrow techniques from the previous protocol. Binding key setup to the selective disclosure proofs helped to protect against NoMatch and Combine events, and thus provided authenticity for the protocol. We will again use the idea of deriving the session key and the binder from the same seed.

To construct a new protocol, we will employ two techniques. First, we define a new type of public-key certificate based on ABCs. Second, in order for the two

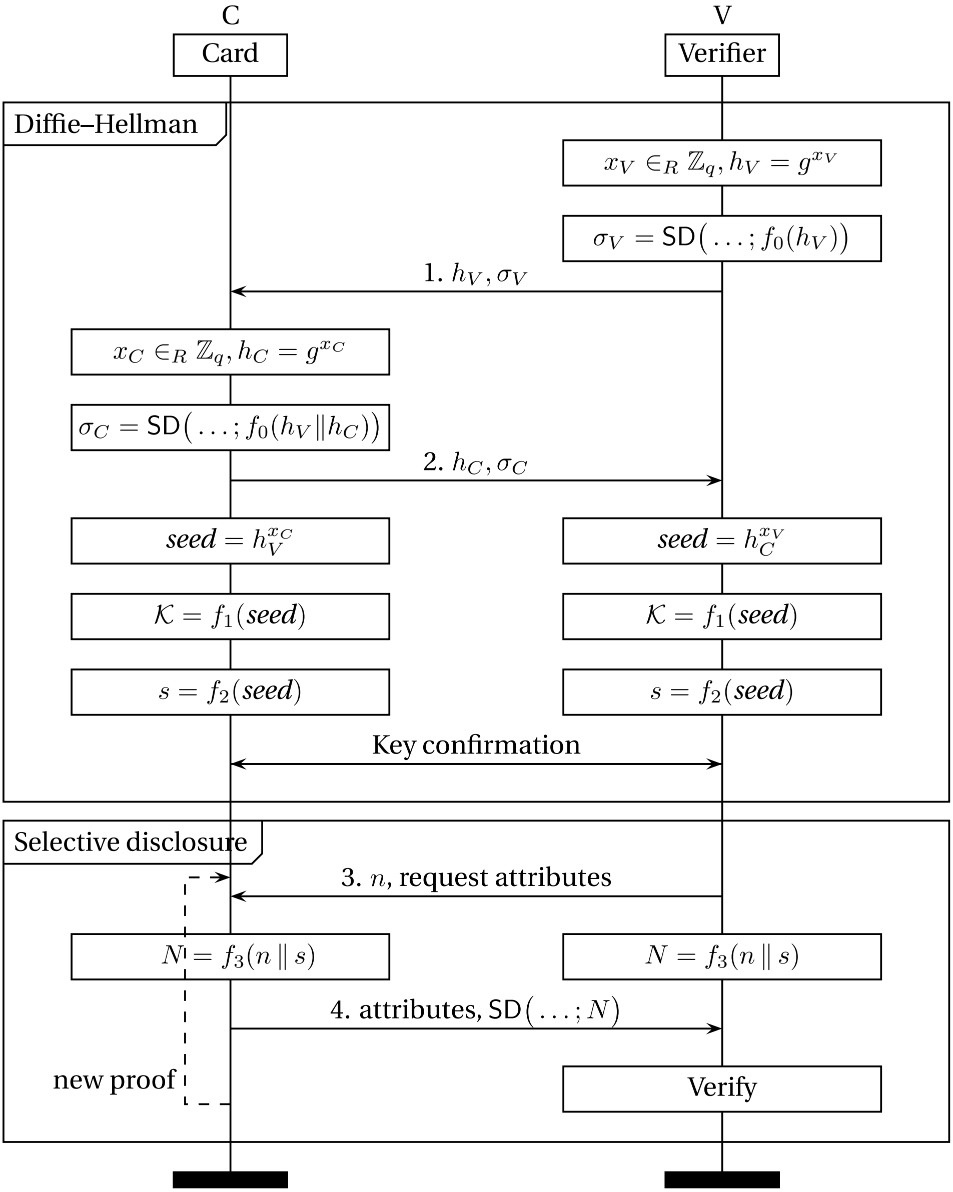

C
Card
V
Verifier
Diffie–Hellman
$x_V \in_R \mathbb{Z}_q, h_V = g^{x_V}$
$\sigma_V = \mathsf{SD}\big(\ldots; f_0(h_V)\big)$
1. h_V, σ_V
$x_C \in_R \mathbb{Z}_q, h_C = g^{x_C}$
$\sigma_C = \mathsf{SD}\big(\ldots; f_0(h_V \| h_C)\big)$
2. h_C, σ_C
$\mathit{seed} = h_V^{x_C}$
$\mathit{seed} = h_C^{x_V}$
$\mathcal{K} = f_1(\mathit{seed})$
$\mathcal{K} = f_1(\mathit{seed})$
$s = f_2(\mathit{seed})$
$s = f_2(\mathit{seed})$
Key confirmation
Selective disclosure
3. n, request attributes
$N = f_3(n \| s)$
$N = f_3(n \| s)$
4. attributes, $\mathsf{SD}\big(\ldots; N\big)$
new proof
Verify

Figure 4.2: ABCDH protocol: ABC authentication and the Diffie–Hellman key exchange for selective disclosure.

parties to set up a session key, they use authenticated Diffie–Hellman (DH) key exchange. Both parties will authenticate using zero-knowledge proofs.

As mentioned above, we define first a new type of certificate, the ***attribute-based public-key credential*** (ABPKC). Such a credential plays a similar role as a conventional public-key certificate, but technically it is an Attribute-Based Credential. A CA issues ABPKCs containing at least the following attributes: (1) the verifier's public key and (2) its access rights `Att_Acc` – encoded as an attribute – to particular attributes in ABCs on smart cards. As any other ABC, an ABPKC also provides the selective disclosure functionality. For various types of use cases a verifier may have various attributes in its ABPKC. Consequently, a verifier can reveal its access rights adaptively depending on use cases. For instance, it is not necessary for a user to know that a service provider is eligible to request the '`gender`' attribute, when the current application needs only the '`over 12`' attribute.

It is well known that the textbook Diffie–Hellman (DH) key-exchange protocol [DH76] is susceptible to man-in-the-middle attacks. The reason for that is that there is nothing that binds the ephemeral public shares to the two parties who intend to establish a session key. While we do not identify cards and verifiers, we are still able to authenticate them and their messages. Each public share is signed by a selective disclosure, that is, by a non-interactive attribute-based zero-knowledge proof of knowledge; see Figure 4.2. We assume that a DH group is given (*e.g.* $\mathbb{G} = \mathbb{Z}_p^*$, a prime subgroup $\langle g \rangle$ of $\mathbb{G}$ of order q, or alternatively q points generated by a point P on an elliptic curve over a finite field) and the participants know all system parameters. Though we use multiplicative notation, the given protocol can easily be adapted to additive groups.

As with the ***ICA*** protocol in Section 4.4, we assume some underlying secure cryptographic primitives, such as random oracles (implemented as standard hash functions), a symmetric-key (authenticated) encryption for the secure channel and the ABC technology that supports selective disclosure of attributes. Notations are similar to those in the previous section.

Protocol Description

The verifier generates his own secret DH share $x_V \in_R \mathbb{Z}_q$ and computes $h_V = g^{x_V}$ in $\mathbb{G}$. Using his ABPKC, he creates a (non-interactive) selective disclosure proof $\sigma_V = \mathsf{SD}\big((pk_V, \texttt{Att\_Acc}); f_0(h_V)\big)$ about attributes that describe his public key pk_V and his attribute access rights `Att_Acc`. The verifier uses $f_0(h_V)$ as the message to be signed by the non-interactive proof. Finally, he sends his DH public share h_V and the proof σ_V to the card.

Upon receiving the verifier's authentication and the public share, the card verifies the proof and the values, and stores the access rights `Att_Acc` – and possibly the public key pk_V as the verifier's identifier although it is not essential for this protocol. It generates its own private share $x_C \in_R \mathbb{Z}_q$ and computes $h_C = g^{x_C}$ in $\mathbb{G}$. The card creates an empty proof of validity (with its root credential) using $(h_V \| h_C)$ as the message to be signed: $\sigma_C = \mathsf{SD}\big(\emptyset; f_0(h_V \| h_C)\big)$. Finally, the card sends its public share h_C and the proof σ_C to the verifier.

The verifier checks the proof. In particular, he checks that the proof σ_C is correct and the message it signs does correspond to h_V he sent previously.

According to the Diffie–Hellman key exchange, both parties can now calculate $\mathit{seed} = g^{x_V x_C}$ and can derive their shared key $\mathcal{K} = f_1(\mathit{seed})$. They can perform an explicit key confirmation. The verifier and the card carry out all the ABC selective disclosure proofs within the secure channel protected by key $\mathcal{K}$. Like in ***ICA***, nonces for the proofs are transformed using $s = f_2(\mathit{seed})$ before they are signed by the NIZK proofs.

Theorem 4.15. *Given a group in which the Computational Diffie–Hellman (CDH) holds and the group description is known to all participants. Further, given a secure ABC technology for selective disclosure with ideal revocation and a CPA secure symmetric-key encryption is given for the secure channel. Moreover, let f_0, f_1, f_2, f_3 be random oracles with all participants (including the adversary) having access to them and assume that for the nonce space holds the following: $|\mathcal{N}| \geq 2k$. Then the **ABCDH** protocol is a secure ABC session protocol.*

Proof. The proof is very similar to the proof of Theorem 4.14, with the following modifications. Clearly, two parties following the protocol can both `Accept` under a benign adversary. Furthermore, the verifier receives all eligibly requested attributes that it can output at the end of the protocol.

1. The adversary does not have a valid verifier certificate. Thus, the only chance for her besides faithfully relaying a verifier's fresh message is to replay an earlier message.

2. Similarly, the adversary cannot produce a fresh reply without a valid card.

3. After the second message, the adversary learns two public DH shares and both of them are authenticated by selective disclosure proofs showing validity of each participant. Relying on the CDH assumption, the adversary cannot compute *seed* and $\mathcal{K}$.

4. After the explicit key confirmation both parties can output `KeyOK`.

5. Again, because of the CDH problem, s can only be computed by the two oracles in a matching conversation. An adversary has negligible probability in the binding game for exactly the same reason as in ***ICA***.

Therefore, ***ABCDH*** is confidential and authentic, and ultimately a secure ABC authentication protocol. □

In this model the verifier is still trusted, that is, the adversary cannot corrupt it. However, in practice if this scheme is implemented, both participants have to have access to the revocation list; so, in case the verifier's ABPKC is revoked, the other party should not accept the verifier's selective disclosure anymore.

Although the ***ABCDH*** protocol is not as efficient as the ***ICA***, it is worth discussing it for multiple reasons. It demonstrates that on an abstract level the verifier and the prover (the card) can be regarded in a symmetric manner. Both of them have an ABC to prove validity and potentially to protect their privacy. In some practical scenarios the verifier may not need to reveal its public key. Second, this enables us to foresee applications that have not been considered yet. Examples include machine-to-machine communication in the internet of things or ad-hoc communication between individuals who do not trust each other and thus wish to share as little information as possible in a given context. As devices become more powerful in terms of storage and computation, this protocol becomes more feasible. Third, the use of ABCs as a new kind of public-key certificate is an innovative approach that can boost the evolution of PKIs and their applications.

4.6 Related Techniques

In this section we make a brief overview of proposals to privacy-friendly authentication and key establishment.

4.6.1 Pseudonyms

Pseudonyms have been proposed in [Cha85], further investigated in [LRSW00], and realised by the Camenisch–Lysyanskaya signature [CL01] and Idemix [IBM12]. The European ABC4Trust project [CKL$^+$11] aims to incorporate pseudonyms with other attribute-based credential systems, such as U-Prove [Paq11].

As a user's master secret key α is included in all credentials, it can play an important role in each proof. A pseudonym is a randomised commitment to α that a user can create. A pseudonym is bound to the credentials, and thus to the user as well, but it does not reveal anything about the key or about the user's identity. Once it is revealed, it cannot be changed during a selective disclosure protocol, preventing users from abusing their credentials. (For instance, mischievous users could combine attribute proofs related to different users.) Therefore, a pseudonym is similar to a public key within a proof session, but there are practically an infinite number of pseudonyms corresponding to a secret key, which allow for unlinkability among separate sessions. Applying pseudonyms, however, requires equality proofs, that is, zero-knowledge proofs that the same secret key α was used in the pseudonym and in the selective disclosure proofs. A selective disclosure and credential issuance require a secure channel that can be established in a more straightforward way if pseudonyms are allowed in a particular implementation. Nevertheless, care has to be taken to bind key establishment and proofs. We propose therefore that implementers of such systems verify that they can achieve a secure ABC authentication protocol in our model.

4.6.2 German eID Model

The German e-identity project [PWVT12] enables citizens to provide privacy-friendly proofs based on attributes, such as '`over 18`' age proofs. However, it employs a different approach to achieve anonymity for particular identity cards. Each card has a public key (so-called chip authentication key) that enables authentication or channel establishment. A public key is not assigned to a single card, but a batch of cards. Therefore, batches can be identified, but not cards (or card holders). To achieve an appropriate level of anonymity, batches should not be too small. On the other hand, too big batch sizes result in organisational problems in case of a card with a corresponding private key gets compromised. In this case not only this particular card, but all cards in the batch have to be revoked.

In the German eID technology the chip-authentication key is not directly applied in the key-establishment phase as it uses a password-based key-agreement protocol, the so-called PACE. This protocol, assuming that the card and the verifier share the card owner's password, enables the derivation of a high-entropy session key from a low-entropy password. Bender et al. [BFK09] give a security proof that the protocol is secure under standard and "close-to-standard" assumptions. (See further considerations about the German eID in Section 6.7.3.)

4.6.3 Anonymous Authentication

There are several techniques proposed for anonymous authentication. They are in general aimed to allow each member of a group to prove that they are indeed members without letting a verifier know their identity. The general security model is that an attacker should not be able to distinguish proofs provided by different members of the group. A possible realisation of these requirements is a group signature, first proposed by Chaum and van Heyst [CVH91]. Ever since their introduction group signatures have been an active field of research. A similar method with slightly different security requirements (*e.g.* more protected anonymity for members) are ring signatures, described by Rivest *et al.* [RST01]. A third approach is anonymous authentication presented by Lindell [Lin07] uses a conventional encryption scheme and requires several dummy encryptions by the verifier.

Membership in groups can be described as special cases of attributes since a group can be identified by a particular attribute. Therefore, ABCs and thus our schemes can be considered to provide a more general approach.

4.6.4 General Credential-Based Authentication

In the context of trusted computing, Camenisch *et al.* propose the Direct Anonymous Attestation (DAA) [BCC04] (standardised) technique. Using it, a trusted platform module within a host device can interact with its host and a remote computer to produce an anonymous proof about the authenticity of the host computer. DAA uses the CL signature similarly to the Idemix technology described in Chapter 3.

The CAID and CAKE (credential-authenticated identification and key exchange) protocols by Camenisch *et al.* [CCGS10] are introduced for authentication and key exchange using credentials. These protocols are proven to be secure in the universal composability framework. Although the motivation and the results are closely related to ours, the present study assumes little about the resources of users' devices (*e.g.* very simple policy, limited computational capability, minimal infrastructure). Therefore, those results cannot be applied directly for our requirements, although further research may suggest adaptability.

4.7 Conclusion

Selective disclosure and dependent credential issuance (*i.e.* one based on already existing credentials; see 6.4.1) are important mechanisms of attribute-based credentials to provide security and privacy simultaneously. ABCs are building blocks of future privacy-friendly electronic identity systems. This chapter showed how to build a secure channel for the selective disclosure mechanism between a verifier and a possibly anonymous smart card carrying ABCs in the presence of a very powerful adversary.

First, we have described a security model to enable us to make security proofs with standard cryptographic primitives and assumptions. Second, we have shown two protocols that are secure in this framework. One of these protocols is more efficient, the other one can be generalised to new scenarios in which devices (a prover and a verifier) authenticate each other anonymously in a symmetric fashion.

We assumed that proper revocation mechanisms exist that can handle abuses of ABCs. Although there exist cryptographic techniques for revocation, most of them are not efficient enough for smart-card implementations. Feasible and easily applicable privacy-preserving revocation techniques are crucial in the deployment of ABCs, but they are yet to be developed (see references and a new approach in Section 6.6.1).

Authentication has been considered as a general notion. Rather than simply a proof of identity, authentication is a proof that certain predicates hold for an entity. When a secure channel is built on this notion of mutual authentication, participants can be convinced that the entities at the other end meet some requirements in terms of these predicates. Privacy-respecting applications will need security analyses in a similar model as the one shown in this chapter. As we mentioned, ***ABCDH*** can be the first step towards many new such protocols.

Chapter 5

Designated Verifier Proofs

In this chapter we attempt to adapt techniques of Attribute-Based Credentials for the RFID context. There are prototypes that show that (even passive) RFID tags are capable of performing a few point multiplications in an elliptic-curve cryptography (ECC) setting,[69] which is the most demanding operation in the ABC protocols. That gives motivation to exploit some of the beneficial privacy properties of ABCs, U-Prove's selective disclosure proof, in particular.

The communication and computational capacity of RFID tags are much more limited than other devices, and the cryptographic modelling also works differently. This is reflected in the design we propose and also the security and privacy proofs we provide. We follow Vaudenay's model[Vau07] and prove that our scheme provides narrow-strong privacy for the tags, relying on the representation and the Decisional Diffie–Hellman (DDH) assumptions.

So far, we have worked with credentials as signed Pedersen commitments, as we discussed it on page 53. Accordingly, a selective disclosure proof included two conceptual parts: a signature verification and a proof of representation (*i.e.* proof of knowledge of a subset of the exponents in the Pedersen commitment).

In this chapter we will exclude credential signatures and make the following modifications. We assume that each RFID tag is initiated with a Pedersen commitment I, functioning as its identifier, and the tag stores all exponents. Furthermore, identifier I gets stored in a database to which verifiers get access (in fact, the database can be public, although it needs to be authentic). Verification of a tag comprises then the disclosure of I, the confirmation that I is in the database, the disclosure of some attributes and a proof of knowledge of the representation of I. Because of the proposed designated verifier functionality, only a legitimate verifier will be able to receive I, to derive the disclosed attributes and to verify the proof. To sum up, only designated verifiers can confirm authenticity of a tag, and they have to access a authentic database instead of verifying a signature.

[69] Five point multiplications on an RFID tag of the hardware design described in [LBSV10] can be performed in less than 500ms, where the area for the elliptic curve (EC) computations is under 15,000 gates, the frequency is 700KHz, the power is 13.8μW, and the energy for each point multiplication is 1.18μJ.

The proposed technique provides an alternative to setting up a secure channel (discussed in Chapter 4) for an ABC selective disclosure protocol between a prover device and a reader terminal. The designated verifier protocol is integrated within the interactive zero-knowledge proof and does not require additional communication rounds. The described scheme is suitable for simple setups in which each tag has only one 'credential' (*i.e.* commitment), a protocol run requires little communication and no additional infrastructure (*e.g.* PKI) is desired.

5.1 Related work

A zero-knowledge proof [GMR89] gives certainty about a statement's validity. For instance, the Schnorr identification protocol [Sch91] (described in Figure 3.3) proves that an entity knows the secret exponent corresponding to his public identity. This proof technique assumes that the verifier receives the identity before (or during) the interaction. Similarly, in the context of ABCs, a selective disclosure proof also assumes that the verifier knows the disclosed attributes that are to be proven in a credential. That means that any party interacting with the prover or eavesdropping on the communication channel between the prover and the verifier can learn the claim and the proof. Since in the RFID contexts interrogation is direct and prior verifier authentication is usually not possible, this is not desirable. *Designated verifier proofs* are interactive protocols, in fact extended zero-knowledge proofs, in which only a designated verifier can obtain valuable information. Moreover, this technique preserves simulatability of zero-knowledge proofs, and therefore, verifiers cannot convince third parties with regards to the prover about the information they were given.

Restriction of verification has a long history in cryptography. Undeniable signatures were introduced in 1989 by Chaum and van Antwerpen [CVA90] and have been enhanced to zero-knowledge proofs of ownership by Chaum [Cha91]. An undeniable signature cannot be verified without interacting with its signer. Furthermore, during the proving protocol no external parties learn anything about the validity or invalidity of the signature. Jacobsson et al. [JSI96] propose a more general notion, the designation of verification that a statement is true. The idea is that the prover generates a zero-knowledge proof that can only be produced by her and the verifier. Since the verifier knows that he was not the one who created the proof, he becomes convinced about the validity of the statement. However, he cannot convince any third party that the proof was produced by the prover and not by himself. Saeednia et al. [SKM04] improve the notion of designated verifier signatures, in which not only the verifier but anybody can simulate transcripts of valid proof conversations. They also propose an efficient designated verifier signature using the Fiat–Shamir heuristic [FS87]. As in all these techniques, also in our scheme the prover is in control over who can verify her proofs.

In the context of RFID systems, public-key cryptography (PKC) is essential for achieving user's privacy [Vau06, Vau07]. ECC has typically been the preferred setting for creating PKC-based protocols because of its computational advantage to other techniques (*e.g.* DL in the multiplicative group of a finite field or RSA). Numerous protocols were designed aiming at security and privacy of RFID systems based on the EC point multiplication [BCI08, LBSV10]. Starting from an authentication of a single tag, a number of more complex protocols emerged such as grouping proofs [Jue04, BLS^{+}11] and hierarchical proofs [BSSV12].

Designated verifier proofs in relation to RFID systems were first proposed by Bringer et al. [BCI08]. They present a scheme that alters the Schnorr identification by incorporating the verifier's public key. First, the protocol conveys not only the proof but also the identifier of the prover tag; second, only the designated verifier can learn the proof and the identifier. Thus, it prevents possibly malicious readers and eavesdropping adversaries from discovering the identifiers of RFID tags.[70]

We further extend the scheme for RFID tags to be able to operate with attributes and to perform selective disclosure proofs. More specifically, we generalise the scheme of Bringer et al. [BCI08] by allowing multiple attributes to be included in an identifier. This is achieved by using a (generalised) Pedersen commitment. A prover in a designated verifier selective disclosure proof can reveal any subset of attributes similarly to ABC proofs. However, unlike those proofs, only a designated verifier can extract disclosed attributes and verify the proof.

5.2 Preliminaries

5.2.1 Cryptographic Preliminaries

In Chapter 3 we stated the DL and the representation problems. Here we recall them but with the additive notation in an ECC setting. Let E be an elliptic curve defined over a finite field $\mathbb{F}_{2^k}$. Fields of characteristic 2 are more suitable for hardware implementations, and for RFID tags in particular, than fields of odd characteristic. Nevertheless, ECC protocols conceptually apply to arbitrary fields. Let $(G, +)$ denote a cyclic group of prime order p of points on the curve E, generated by a point P. We use capital letters, like A and P, to denote points on the elliptic curve. Scalars are written using lower case letters. We write kP to denote the point P added k times to itself. Finally, we denote by $x \in_R \mathbb{Z}_p$ if x is chosen uniformly at random from the set $\mathbb{Z}_p$. For more details we refer the reader to *e.g.* [Kob87, HWF09].

The discrete logarithm problem (Definition 3.2 on page 44) and the representation problem (Definition 3.6 on page 45) with EC notation are as follows:

Definition 5.1. The ***EC discrete logarithm problem*** (ECDL problem) takes the following form: Given $P \in G$ and $Q := xP \in G$, find x.

[70] Since Bringer et al.'s technique randomises the the messages that a tag sends from external parties' point of view, they call their scheme the "Randomized Schnorr".

Definition 5.2. Given an EC group G and elements $P_1, \dots, P_L \in G$. Then $(x_1, \dots, x_L) \in \mathbb{Z}_p^L$ is said to be an ***EC representation*** of Q if $Q = \sum_{i=1}^{L} x_i P_i$ with respect to $(P_1, \dots, P_L)$.

The ***EC representation problem*** takes the following form: Given $P \in G$ and $Q = \sum_{i=1}^{L} x_i P_i \in G$, find a tuple $(x'_1, \dots, x'_L) \in \mathbb{Z}_p^L$ for which $Q = \sum_{i=1}^{L} x'_i P_i$. (We omit the adjective 'EC' whenever it is clear from the context.)

Both the ECDL problem and the EC representation problem are assumed to be hard (*i.e.* no polynomially bounded adversary can solve these problems). We note that while the DL problem has a unique solution, the representation problem has plenty of different ones (in fact, p^{L-1} *cf.* footnote 52).

We also need an additional assumption about the difficulty of the following problem.

Definition 5.3. The ***Decisional Diffie–Hellman problem*** (DDH problem) takes the following form: Given a generator P, and the points $A = aP$, $B = bP$ and $C = cP$, where $a, b \in_R \mathbb{Z}_p$, determine whether $c \equiv ab \pmod{p}$.

5.2.2 Selective disclosure revisited

A selective disclosure protocol enables a prover (or user) to show some of the attributes from her ABC, or here from her Pedersen commitment, and to prove the rest of the attributes in a zero-knowledge fashion. The disclosing set determines which attributes are revealed. Just to recap: A ***disclosing set*** $\mathcal{D} \subseteq \{1, \dots, L\}$ is a set of indices of the disclosed attributes.

As we mentioned above, a U-Prove selective disclosure scheme in fact comprises two steps: (1) a signature verification; (2) a proof of representation. Here we only focus on step (2), since U-Prove's credential signature is not applied in this setting; rather the verifier checks whether I is in the trusted system database DB of valid identifiers. The proof of representation is concerned with those attributes that are not revealed. All disclosed attributes have to be sent through the communication channel to the verifier. To make this explicit and to adapt the protocol to the RFID context, we have made some modifications in the description of the scheme in Figure 5.1 in comparison with that in Figure 3.6. We use ECC notation; we include the identifier and the disclosed attributes in the response phase rather than in the common input; and we re-write the verification equation to express the identifier I (corresponding to h' in U-Prove). These adjustments make it clearer to extend the protocol to our designated verifier proof while they do not affect the security properties. Hereafter, we refer to the proof of a representation as 'U-Prove selective disclosure'.

It is clear that the identifier as well as all revealed attributes are easily accessible for an eavesdropper if we cannot presume an underlying secure channel. Also, a rogue verifier terminal could interrogate the prover and receive this information. Our goal is to prevent the possibility of these malicious scenarios.

Prover		**Verifier**
$x_0, \ldots, x_L$ $I = \sum_{i=0}^{L} x_i P_i$	$P_0, \ldots, P_L$	
$\forall i \notin \mathcal{D} : \alpha_i \in_R \mathbb{Z}_p$		
$A := \sum_{i \notin \mathcal{D}} \alpha_i P_i$	$\xrightarrow{A}$	
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_p$
$\forall i \notin \mathcal{D}$:		
$r_i := c \cdot x_i + \alpha_i \pmod{p}$	$\xrightarrow{I, (x_i)_{i \in \mathcal{D}}, (r_i)_{i \notin \mathcal{D}}}$	Verify the identity and disclosed attributes: $I \overset{?}{=} c^{-1}(\sum_{i \notin \mathcal{D}} r_i P_i - A) + \sum_{i \in \mathcal{D}} x_i P_i$

Figure 5.1: Adapted version of U-Prove's selective disclosure scheme; *cf.* Figure 3.6. (We mention that there are $L + 1$ point multiplications and that the Schnorr identification is a special case of this scheme where $L = 0$ and $\mathcal{D} = \emptyset$.)

5.3 Designated verifier selective disclosure

5.3.1 Participants

A system includes an issuer, RFID tags and designated verifiers. An ***issuer*** sets up a system and initiates RFID ***tag***s. Each time when a tag has been set up, its identifier is stored in a database DB and the corresponding secret values are stored on the tag. When a tag is interrogated by a ***designated verifier***, a selective disclosure protocol takes place. The protocol is zero-knowledge so that the verifier does not learn anything except the identifier of the tag. After the interaction, the verifier looks up the resulting identifier in DB and considers the tag valid if it can be found there. An external party (an adversary or an illegitimate verifier) does not acquire any valuable knowledge from the interaction. We mainly focus here on authentication protocols and not issuing.

5.3.2 Designated Verifier Identification (DID)

First we briefly describe an intermediary scheme that we call the designated verifier identification (DID) scheme. It is a simple generalisation of Bringer et al.'s "Randomized Schnorr" [BCI08] scheme (in which $L = 0$), similar to how the DL problem generalises to the representation problem; see Definition 3.6 on page 45. In its pure form this protocol should not be applied since it is less efficient than the "Randomized Schnorr", but achieves the same identification. (Essentially, this is an 'empty proof' in which no attributes are revealed, only the fact is proven that the representation of I is known to the prover.) Yet it is a stepping stone to the extension of this scheme in which a tag can reveal not only its identifier but also any subset of its attributes to a designated verifier.

The designated verifier identification scheme allows a prover to prove the knowledge of a representation and to reveal its identifier I to a designated verifier with public key V. Verification and the computation of I can only be performed by a

Prover $x_0, \dots, x_L$, $I = \sum_{i=0}^{L} x_i P_i$	$P_0, \dots, P_L$ $V = v \cdot \sum_{i=0}^{L} P_i$	**Verifier** v
$\alpha_0, \alpha_1, \dots, \alpha_L, \beta \in_R \mathbb{Z}_p$ $A_1 := \sum_{i=0}^{L} \alpha_i P_i$		
$A_2 := \beta V$	$\xrightarrow{A_1, A_2}$	
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_p$
$\forall i \in 0, \dots, L$: $r_i := c \cdot x_i + \alpha_i + \beta \pmod{p}$	$\xrightarrow{r_1 \dots r_L}$	Compute the identifier and verify its validity: $I = c^{-1}(\sum_{i=0}^{L} r_i P_i - A_1 - v^{-1} A_2)$

Figure 5.2: The designated verifier identification (DID) protocol. The Randomized Schnorr scheme is a special case, for which $L = 0$. (There are $L + 2$ point multiplications on the prover's side.)

legitimate verifier, that is, one that is privy to the corresponding secret key v. The scheme has the following algorithms:

- $\mathsf{Gen}^{\mathcal{I}}(k)$ outputs Sys (depending on security parameter k) such that $Sys = (k, G, P_0 = P, P_1 = p_1 P, \dots, P_L = p_L P)$ (where P is a generator in an ECC group G and $p_1, \dots, p_L \in \mathbb{Z}_p$). The scalars $p_1, \dots, p_L$ are erased.

- $\mathsf{Gen}^{\mathcal{P}}(Sys)$ outputs $(I, sk^{\mathcal{P}})$ such that $I = \sum_{i=0}^{L} x_i P_i$ and $sk^{\mathcal{P}} = (x_0, \dots, x_L) \in \mathbb{Z}_p^{L+1}$. Additionally, I is stored in database DB of valid prover identifiers. (Only the issuer is assumed to have write access to this database.)

- $\mathsf{Gen}^{\mathcal{V}}(Sys)$ outputs $(pk^{\mathcal{V}}, sk^{\mathcal{V}})$ such that $sk^{\mathcal{V}} = v$ and $pk^{\mathcal{V}} = V = v \cdot \sum_{i=0}^{L} P_i$. The verifier's public key V is published and available for all participants in the system.

- The DID protocol is shown in Figure 5.2

This protocol is a modified version of an interactive zero-knowledge proof of knowledge of a representation of I with respect to the generators $(P_0, \dots, P_L)$. Two changes are required on the prover's side and one on the verifier's side. The prover needs an additional commitment A_2 to a random value β using the verifier's public key V. This random value β is also added to each responses. The knowledge of the secret key v makes it possible for the designated verifier (only) to cancel $\beta \sum_{i=0}^{L} P_i$ and eventually compute the prover's identifier. The protocol is *complete* – *i.e.* an honest prover can convince a legitimate verifier – because of the simple fact that

$$\begin{aligned}
A_1 + v^{-1}A_2 &= \sum_{i=0}^{L} \alpha_i P_i + v^{-1}\beta V \\
&= \sum_{i=0}^{L} \alpha_i P_i + v^{-1}\beta v \cdot \sum_{i=0}^{L} P_i \\
&= \sum_{i=0}^{L} \alpha_i P_i + \beta \cdot \sum_{i=0}^{L} P_i \\
&= \sum_{i=0}^{L} (\alpha_i + \beta) P_i;
\end{aligned}$$

and therefore, the verifier can indeed compute the identifier I:

$$\begin{aligned}
c^{-1}\left(\sum_{i=0}^{L} r_i P_i - A_1 - v^{-1}A_2\right) &= c^{-1}\left(\sum_{i=0}^{L}(c \cdot x_i + \alpha_i + \beta)P_i - \sum_{i=0}^{L}(\alpha_i + \beta)P_i\right) \\
&= c^{-1}\left(\sum_{i=0}^{L} c x_i P_i\right) = \sum_{i=0}^{L} x_i P_i \\
&= I.
\end{aligned}$$

5.3.3 Designated verifier selective disclosure (DSD)

We are ready to describe the designated verifier selective disclosure (DSD) protocol. We need two new concepts: an entitlement set and an attribute point.

Definition 5.4. An ***entitlement set*** $\mathcal{E} \subseteq \{1, \ldots, L\}$ is a set of indices of attributes that a verifier can legitimately extract from tags. A verifier has the secret keys corresponding to the indices from his entitlement set.

Note that an entitlement set places a comparable restriction on the verifier as the attribute access policy in a public-key certificate in Chapter 4.

Let us introduce the notion of ***attribute point***s. In the DSD scheme, defined below, verifiers can compute attribute points rather than their values directly. The issuer creates a public look-up table of possible attribute values $x_i \in \mathbb{Z}_P$ and points $C_i = x_i P_i$. Thus, this table is available for all verifiers. We illustrate this with a mock table of the attribute '`blood type`' with attribute values and attribute points (sorted by the attribute value here for readability):

Prover		**Verifier**
$x_0, \dots, x_L$ $I = \sum_{i=0}^{L} x_i P_i$	$P_0, \dots, P_L$ $\forall i \in \mathcal{D}: \; V_i = v_i P_i$ $V = v \cdot \sum_{i=0}^{L} P_i$	$v, (v_i)_{i \in \mathcal{E}}$
$\alpha_0, \dots, \alpha_L, \beta \in_R \mathbb{Z}_p$ $A_1 := \sum_{i=0}^{L} \alpha_i P_i$ $A_2 := \beta V$ $B_i := (\alpha_i + \beta) V_i \quad \forall i \in \mathcal{D}$	$\xrightarrow{A_1, A_2, (B_i)_{i \in \mathcal{D}}}$	
	$\xleftarrow{c}$	$c \in_R \mathbb{Z}_p$
$\forall i \in 0, \dots, L$: $r_i := c \cdot x_i + \alpha_i + \beta \pmod{p}$	$\xrightarrow{r_0 \cdots r_L}$	First verify that the identifier is correct: $I := c^{-1}(\sum_{i=0}^{L} r_i P_i - A_1 - v^{-1} A_2)$ Then for each $j \in \mathcal{D} \cap \mathcal{E}$ compute attribute point C_j: $C_j := (c v_j)^{-1} (r_j V_j - B_j)$

Figure 5.3: The designated verifier selective disclosure (DSD) protocol in which attributes in $\mathcal{D}$ are disclosed. (There are $L + 2 + |\mathcal{D}|$ point multiplications on the prover's side.)

blood type

attribute value	C
A+	`0x23D...A0`
A-	`0xEAA...2B`
B+	`0x3CC...D1`
...	
0-	`0x170...66`

The ***DSD*** scheme cryptographically enforces verifiers to receive at most those attributes that they are entitled to.

- $\mathsf{Gen}^{\mathcal{I}}(k)$ outputs Sys such that $Sys = (k, P_0 = P, P_1 = p_1 P, \dots, P_L = p_L P, (\overline{\mathcal{E}^{\mathcal{V}}}))$ (where P is a generator in an ECC group G, $p_1, \dots, p_L \in \mathbb{Z}_p$, and $(\overline{\mathcal{E}^{\mathcal{V}}})$ is the set of all verifiers' entitlement sets). The scalars $p_1, \dots, p_L$ are erased.

- $\mathsf{Gen}^{\mathcal{P}}(Sys)$ outputs $(I, sk^{\mathcal{P}})$ such that $I = \sum_{i=0}^{L} x_i P_i$ and $sk^{\mathcal{P}} = (x_0, \dots, x_L) \in \mathbb{Z}_p^{L+1}$. The identifier I gets stored in DB. (Again, only the issuer is assumed to have write access to database DB.)

- $\mathsf{Gen}^{\mathcal{V}}(Sys)$ outputs $(pk^{\mathcal{V}}, sk^{\mathcal{V}})$ such that $sk^{\mathcal{V}} = (v, (v_i)_{i \in \mathcal{E}})$ (elements of $\mathbb{Z}_p$) and $pk^{\mathcal{V}} = (V, (V_i)_{i \in \mathcal{E}})$ where $V = v \cdot \sum_{i=0}^{L} P_i$ and $V_i = v_i P_i$. The verifier's public key $(V, (V_i)_{i \in \mathcal{E}})$ is published and available for all participants in the system.

- The DSD protocol is shown in Figure 5.3

This protocol is an extension of the DID protocol; it is not only a proof of knowledge of a representation, but also the selective disclosure of any subset of the attributes. The idea behind a designated selective disclosure is that the capability of computing disclosed attributes is not 'all-or-nothing', it is determined *independently for each attribute.* A public key V is used to designate the identifier, and separate public keys V_i are used to designate each attribute i. For instance, it can happen that a verifier is entitled to receive a tag's identifier and the first attribute (*e.g.* the '`blood type`' above), but not entitled to the rest of the attributes (*e.g.* other medical data), even if the tag disclosed all attributes.

In comparison with the DID scheme, two changes are required, one on the prover's and one on the verifier's side. The tag computes B_i for each disclosed attributes in the commitment phase. The verifier computes the attribute points C_i, based on which he can retrieve the actual attribute values (from the attribute look-up table).

The identifier I computed in the same way as in DID. The selectively disclosed attribute points can correctly be computed:

$$\begin{aligned}(cv_j)^{-1}(r_jV_j - B_j) &= (cv_j)^{-1}((cx_j + \alpha_j + \beta)V_j - (\alpha_j + \beta)V_j) \\ &= (cv_j)^{-1}cx_jV_j = x_jP_j \\ &= C_j.\end{aligned}$$

5.4 The security of DSD

As we see, the DSD protocol is complete, that is, a valid tag can reveal its identifier and attributes to a verifier. In this subsection we discuss the security of the protocol. First we show that the scheme is zero knowledge, that is, the verifier does not lear any additional information except the identifier and those attributes that are revealed and are in his entitlement set. Second, using Vaudenay's model we prove that the DSD protocol is secure and provides narrow-strong privacy.

5.4.1 Zero-knowledge

Theorem 5.5. *The designated verifier selective disclosure (DSD) protocol is (honest-verifier) perfect zero knowledge.*

Proof. We show that an honest, designated verifier does not learn any secret value from a prover, only the fact that it has a valid identifier (I) with any disclosed attributes (C_i). We assume that $\mathcal{D} = \mathcal{E}$.

As usual, we prove that the verifier himself could simulate valid proof transcripts. Simulated conversations take also the designated verifier's private keys as input.

Real conversation:

- **Input:** $c \in \mathbb{Z}_p, x_0, x_1, \ldots, x_L, P_0, \ldots, P_L, V, \mathcal{D}, \forall_{i \in \mathcal{D}} : V_i$.
- **Simulation:**
  1. Choose random $\alpha_0, \ldots, \alpha_L, \beta \in_R \mathbb{Z}_p$
  2. Set $A_1 := \sum_{i=0}^{L} \alpha_i P_i, A_2 := \beta V, \forall_{i \in \mathcal{D}} : B_i := (\alpha_i + \beta) V_i$
  3. Set $\forall_{i \in 0,\ldots,L} : r_i := c \cdot x_i + \alpha_i + \beta \pmod{p}$
- **Output:** $(A_1, A_2, \forall_{i \in \mathcal{D}} : B_i, c, (r_0, \ldots, r_L))$

Simulated conversation:

- **Input:** $c \in \mathbb{Z}_p, I, P_0, \ldots, P_L, V, \forall_{i \in \mathcal{E}} : (v_i, V_i)$ and any C_i.
- **Simulation:**
  1. Choose $r_0, \ldots, r_L \in_R \mathbb{Z}_p$
  2. Compute $A := \left(\sum_{i=0}^{L} r_i P_i\right) - cI$
  3. Compute $B_i := r_i V_i - c v_i C_i \ \forall_{i \in \mathcal{E}}$
  4. Choose $\beta \in_R \mathbb{Z}_p$
  5. Compute $A_2 := \beta V$ and $A_1 := A - \beta \sum_{i=0}^{L} P_i$
- **Output:** $(A_1, A_2, \forall_{i \in \mathcal{E}} : B_i, c, (r_0, \ldots, r_L))$

Since $\alpha_0, \ldots, \alpha_L, \beta$ are chosen uniformly at random (in $\mathbb{Z}_p$), r_i's and A_1, A_2 are uniformly distributed in a real execution. In simulated conversations the distribution of r_i's is uniform by definition, thus the distribution of the points B_i is uniform as well. Because of the uniformity of β, A_1, A_2 are also uniformly distributed in a simulated conversation. In sum, given the system parameters and the values to be proven, a verifier (for which $\mathcal{D} = \mathcal{E}$) can compute a tuple that has exactly the same distribution as real conversations. Therefore, the simulation (as well as the zero-knowledgeness) is perfect. □

Interestingly, it is clear from the proof (see Step 3. of the simulation) that a verifier can only simulate proofs with disclosed attributes from his entitlement set, *i.e.* he needs the corresponding secret attribute key.

5.4.2 Security model

Besides being complete and zero-knowledge, a designated verifier scheme also has to satisfy two distinct requirements: security and privacy. The former roughly means that it is impossible for an adversary to impersonate a valid tag, while the latter means that an adversary cannot trace particular tags because it cannot even distinguish legitimate tags from simulated ones. Below we focus on security and privacy in more detail.

Just as Bringer et al. [BCI08], we follow the security model proposed by Vaudenay [Vau07]. In Vaudenay's model privacy is derived from the fact that an adversary is not able to deduce identity information from interaction messages. Also, an adversary cannot trace or identify tags without corrupting them. In his model, Vaudenay describes how adversaries can interact with a set of tags. Besides offering methods for communicating with and choosing from the tags as well as communicating with the reader, the model also exposes two types of oracle calls. The level of access to these oracles defines the type of an adversary.

The first additional oracle is the Result oracle. As it is typical in RFID identification protocols, the reader draws one of the following two conclusions at the end of the protocol. It either concludes that the tag it communicated with has been successfully identified as the tag with identity I, or it reports failure. The Result oracle will return only the success/failure status of the reader. In our protocols we do not allow this type of query. This is called a ***narrow*** adversarial model as opposed to a ***wide*** one, in which the adversary is allowed to make such queries.

The second additional oracle is the Corruption oracle. This allows the adversary to corrupt a tag and as a result learn all its secret values. We consider only ***strong*** attackers, *i.e.* attackers that can obtain the secrets of any tags they choose. In the privacy game, further attacks on the privacy of these tags are allowed afterwards, while they are (of course) explicitly prohibited in the security game.

The active adversary has the power to corrupt tags and run (designated verifier) identification protocols and her goal is to identify successfully (*i.e.* impersonate a tag) to an honest verifier. More specifically, we now give games to define the security model.

Definition 5.6 (***Security Game***)**.** Assume that there exists a system of t tags that can be interrogated via the authentication protocol, then the game consists of two phases:

1. In the first phase, the adversary is allowed to interrogate any tag multiple times. Furthermore, she is allowed to corrupt any tags of her choosing. (The number of queries is bounded polynomially by the security parameter k.)

2. In the second phase, the adversary communicates with the verifier to impersonate one of the uncorrupted tags of the system.

A designated verifier identification scheme is ***secure*** if no adversary can win the Security game above with non-negligible advantage.

Intuitively, the notion of privacy for these types of RFID protocols means that it is not possible to link two different instances of the protocol involving the same tag. This property is often referred to as ***unlinkability***. In Vaudenay's model it is captured as follows. Even though the adversary is given the identifiers of the tags she interrogated at the end of the query phase of the privacy game, she cannot distinguish between the setting in which it communicates with actual tags and the setting in which she communicates with simulated tags. Note that in the latter case the simulator does not know the identifiers. Hence, any information leak on the identifiers can be used by the adversary to gain advantage. A system has ***narrow-strong privacy*** if no adversary can win the following game against a challenger with non-negligible probability.

Definition 5.7 (***Narrow-Strong Privacy Game***)**.** Assume that there exists a system of t tags that can be interrogated via the designated verifier identification protocol. First, the challenger chooses a random bit $b \in_R \{0,1\}$ and depending on it, he runs different experiments:

- If $b = 0$, the adversary is allowed to directly talk to any tag of her choice.
- If $b = 1$, the adversary is not allowed to interrogate tags directly but the challenger, without interacting with the actual tags, simulates them.

Then in the corruption phase, the adversary can receive all private information of the tags by corrupting them. At the end of the game, the adversary must guess the value of bit b.

Note that if the adversary cannot distinguish real communications from simulated ones even when knowing all secret and public information of the tag, she is also unable to trace specific tags. This explains the notion of privacy in this setting.

5.4.3 Security

To prove the security of the DSD scheme, we will use the description of U-Prove's selective disclosure given in Figure 5.1.

Theorem 5.8. *Assuming that U-Prove's selective disclosure scheme is secure against active impersonation attacks the designated verifier selective disclosure (DSD) scheme is also secure against active impersonation attacks.*

Proof. We show how an adversary against the DSD can be used to break the security of U-Prove's selective disclosure scheme. To do so, we build an adversary $\mathcal{B}$ that essentially translates between these two systems. Suppose we are given an adversary $\mathcal{A}$ that wins the active impersonation game against the DSD scheme. We show how to construct an adversary $\mathcal{B}$ that wins the active impersonation game for the U-Prove selective disclosure scheme. This means that $\mathcal{B}$ can produce a valid (one of the identifiers from the database), previously not corrupted identifier with a proof of representation with respect to the base points.

Figure 5.4 shows the attack. We need to show how $\mathcal{B}$ converts messages in both phases. First, in the query phase it answers the identification requests from $\mathcal{A}$ using only the U-Prove oracle acting as tags. Second, $\mathcal{B}$ converts the impersonation attack by $\mathcal{A}$ against the DSD scheme into an identification protocol for the U-Prove selective disclosure scheme. In the latter phase the U-Prove oracle acts as an (honest) verifier. We assume that $\mathcal{B}$ has access to the public databases of (1) valid identifiers and (2) the pairs of attribute values and attribute points.

Initially, adversary $\mathcal{B}$ generates a random private key v and sets the public key V to $V = v\sum P_i$. Furthermore, for any entitled attributes $i \in \mathcal{E}$ adversary $\mathcal{B}$ generates v_i at random and sets $V_i = v_i P_i$, and sends these V_i's together with V to adversary $\mathcal{A}$. (For clarity, we set $\mathcal{D} = \mathcal{E}$. In both phases both systems, U-Prove as well as $\mathcal{A}$, have the same access to disclosed attributes.)

U-Prove system	$P_0, \dots, P_L, \mathcal{D}$	$\mathcal{B}$	$P_0, \dots, P_L, \mathcal{D}$	$\mathcal{A}$
		Initialisation		
		$v \in_R \mathbb{Z}_p, V := v\sum_{i=0}^{L} P$		
		$\forall i \in \mathcal{D} : v_i \in_R \mathbb{Z}_p, V_i := v_i P_i$	$\xrightarrow{V,(V_i)_{i\in\mathcal{D}}}$	
		First phase: Query		
Acting as a tag Selected tag's secret: $x_0, \dots, x_L$ $\forall i \notin \mathcal{D} : \alpha_i \in_R \mathbb{Z}_p$				
$A := \sum_{i\notin\mathcal{D}} \alpha_i P_i$	$\xrightarrow{A}$	$\forall i \in \mathcal{D} : \alpha_i \in_R Z_p, \beta \in_R Z_p$		
		$A_1 := A + \sum_{i\in\mathcal{D}} \alpha_i P_i$		
		$A_2 := \beta V$		
		$\forall i \in \mathcal{D} : B_i := (\alpha_i + \beta)V_i$	$\xrightarrow[(B_i)_{i\in\mathcal{D}}]{A_1, A_2,}$	
	$\xleftarrow{c}$	$\longleftarrow$ Forward c —	$\xleftarrow{c}$	
$\forall i \notin \mathcal{D} : r_i := cx_i + \alpha_i$	$\xrightarrow[(r_i)_{i\notin\mathcal{D}}]{I,(x_i)_{i\in\mathcal{D}},}$	$\forall i \notin \mathcal{D} : r'_i := r_i + \beta$		
		$\forall i \in \mathcal{D} : r'_i := cx_i + \alpha_i + \beta$	$\xrightarrow{r'_0,\dots,r'_L}$	
		Second phase: Impersonation		
Acting as a verifier	$\xleftarrow{A}$	$A := A_1 + v^{-1}A_2$	$\xleftarrow[(B_i)_{i\in\mathcal{D}}]{A_1, A_2,}$	
	$\xrightarrow{c}$	— Forward $c \longrightarrow$	$\xrightarrow{c}$	
		$I := c^{-1}(\sum_{i=0}^{L} r_i P_i - A_1 - v^{-1}A_2)$	$\xleftarrow{r_0,\dots,r_L}$	
		If I valid:		
		$\forall i \in \mathcal{D} : C_i := (cv_i)^{-1}(r_i V_i - B_i)$		
Accept	$\xleftarrow[(r_i)_{i\notin\mathcal{D}}]{I,(x_i)_{i\in\mathcal{D}},}$	Look up x_i corresponding to C_i		

Figure 5.4: The security game in which $\mathcal{B}$ makes use of $\mathcal{A}$ to successfully attack the U-Prove system. $\mathcal{B}$ does not have any insight or influence on the internal states and operations of the U-Prove oracle as well as of adversary $\mathcal{A}$; we denote this fact by shading the corresponding columns. However, we describe the operations the U-Prove oracle is supposed to do in the first phase. (For simplicity we assume $\mathcal{E} = \mathcal{D}$.)

During the first phase $\mathcal{B}$ answers interrogation queries for a tag as follows. First, it queries the U-Prove oracle, who sends a commitment A. Adversary $\mathcal{B}$ generates

$\alpha_i \in_R \mathbb{Z}_p$ for all $i \in \mathcal{D}$, $\beta \in_R \mathbb{Z}_p$, and sends to $\mathcal{A}$ the values

$$\begin{aligned} A_1 &= A + \sum_{i \in \mathcal{D}} \alpha_i P_i \\ A_2 &= \beta V \\ B_i &= (\alpha_i + \beta) V_i \qquad \forall i \in \mathcal{D}. \end{aligned}$$

Subsequently, $\mathcal{B}$ receives challenge c from $\mathcal{A}$ which it passes along to its U-Prove oracle. In return it receives identifier I and r_i for $i \notin \mathcal{D}$ and x_i for $i \in \mathcal{D}$. $\mathcal{B}$ creates and sends responses for $\mathcal{A}$ as follows:

$$r'_i = \begin{cases} r_i + \beta & \text{for } i \notin \mathcal{D} \\ cx_i + \alpha_i + \beta \pmod{p} & \text{for } i \in \mathcal{D}. \end{cases}$$

Clearly, this construction is a perfect simulation of the designated verification protocol for $\mathcal{A}$. This phase can be repeated multiple times (polynomial in the security parameter k).

In the second phase adversary $\mathcal{A}$ can impersonate a tag with non-negligible probability according to the assumption. The goal of adversary $\mathcal{B}$ is to transform this communication such that it in turn impersonates a valid tag for the U-Prove selective disclosure protocol. This means that $\mathcal{B}$ can create a proof of a valid identifier I in the U-Prove system.

First, $\mathcal{A}$ sends two commitments A_1 and A_2, which are converted by $\mathcal{B}$ into $A := A_1 + v^{-1}A_2 \left(= \sum_{i=0}^{L}(\alpha_i + \beta)P_i\right)$ before sending it to the original U-Prove verifier. The verifier responds with a challenge c, which $\mathcal{B}$ relays unchanged to $\mathcal{A}$. Finally, $\mathcal{A}$ replies with the r_i values. $\mathcal{B}$ computes identifier I and checks whether it is valid. If I is not valid, it halts. Otherwise, for $i \notin \mathcal{D}$, $\mathcal{B}$ forwards these values to the challenger. Note that they should equal $r_i = cx_i + \alpha_i + \beta$ and are therefore appropriate responses corresponding to commitment A. For the disclosed attributes ($i \in \mathcal{D}$), $\mathcal{B}$ can calculate

$$C_i \quad = \quad (cv_i)^{-1}(r_i V_i - B_i)$$

Accessing the public database of attribute points and values, *i.e.* (C_i, x_i), $\mathcal{B}$ can then recover the attributes x_i before forwarding them to the challenger according to the U-Prove protocol (see Figure 5.1). This completes the proof. □

5.4.4 Privacy

Theorem 5.9. *Assuming the hardness of the DDH-problem, the designated verifier selective disclosure (DSD) scheme is narrow-strong private.*

Proof. We extend traces for the Randomized Schnorr protocol (the special case of Figure 5.2 when $L = 0$) to full traces for the DSD protocol. We do this in such a way that the new responses are random if and only if the response of the original instance was random. Therefore, any adversary against the DSD scheme can be converted into a Randomized Schnorr adversary. Since the latter is secure under the DDH-assumption [BCI08, Theorem 2], the result follows.

A transcript in our DSD protocol has the form:

$$\left(A_1 = \sum_{i=0}^{L} \alpha_i P_i, \quad A_2 = \beta V, \quad (B_i)_{i\in\mathcal{D}}, \quad c, \quad (r_i = cx_i + \alpha_i + \beta)_{i=0}^{L} \right).$$

Let us show that the adversary cannot tell apart properly constructed r_i's from randomly chosen ones. Following the argument in Bringer et al. [BCI08], we can take out the terms with the secret values of x_i's.[71] Consequently, the adversary has to distinguish instances of the actual distribution

$$D_A^L = \left\{ \left(A_1 = \sum_{i=0}^{L} \alpha_i P_i,\ A_2 = \beta V,\ (B_i)_{i\in\mathcal{D}},\ (r_i = \alpha_i + \beta) \right) \middle| \ \alpha_i, \beta \in_R \mathbb{Z}_p, 0 \le i \le L \right\}$$

from instances of the simulated distribution (with random responses r_i's)

$$D_S^L = \left\{ \left(A_1 = \sum_{i=0}^{L} \alpha_i P_i,\ A_2 = \beta V,\ (B_i)_{i\in\mathcal{D}},\ (r_i) \right) \middle| \ \alpha_i, \beta, r_i \in_R \mathbb{Z}_p, 0 \le i \le L \right\}.$$

Suppose we have an oracle for distinguishing these two distributions. We will use this to decide between the corresponding instances for the Randomized Schnorr scheme, which are in fact instances from D_A^0 or D_S^0. The main idea is that we use a Randomized Schnorr instance $(\widehat{A}_1 := \widehat{\alpha}\widehat{P}, \widehat{A}_2 := \widehat{\beta}\widehat{V}, \widehat{r})$ obtained as a challenge,[72] to construct a full instance for the adversary. Provided that this adversary can distinguish the constructed transcript, we can also break the challenge. The first attribute is reused from the original challenge, *i.e.* $\alpha_0 := \widehat{\alpha}$, and all the other, 'dummy' attributes are computed as $\alpha_i := \widehat{\alpha} + \gamma_i$ where γ_i is random for all $i \in \{1, \ldots, L\}$. All further base points and the extensions of the commitment values are generated similarly with 'dummy' values.

To set up the DSD system for the adversary, we generate the base points $P_0 := \widehat{P}$ and $P_i := p_i\widehat{P}$, with $p_i \in_R \mathbb{Z}_p$ random for all $i \in \{1, \ldots, L\}$. Moreover, we construct the designated public keys V and V_i's:[73]

$$V := \left(1 + \sum_{i=1}^{L} p_i \right) \cdot \widehat{V} \quad \left(= \widehat{v} \cdot \left(\widehat{P} + \sum_{i=1}^{L} P_i \right) = \widehat{v} \cdot \sum_{i=0}^{L} P_i \right).$$

[71] Since the adversary can receive all the secret values x_i of the tags and she knows all the corresponding challenges c, she can make this modification herself.

[72] We denote all the values in the challenge system with a hat (like $\widehat{x}$) no matter if it is secret or not.

[73] In the brackets we explain why we carry out the computations in the given way.

We choose $v_i \in_R \mathbb{Z}_p$ and set $V_i := v_i P_i$ for all $i \in \mathcal{D}$. Finally, we send the adversary the points: $(P_i)_{i=0,\ldots,L}$, V, and $(V_i)_{i\in\mathcal{D}}$.

Next we construct A_1 and A_2 by extending $\widehat{A}_1, \widehat{A}_2$ from the challenge as follows:

$$A_1 := \widehat{A}_1 + \sum_{i=1}^{L} \left(p_i \widehat{A}_1 + \gamma_i P_i\right) \quad \left(= \widehat{\alpha}\widehat{P} + \sum_{i=1}^{L} (\widehat{\alpha} + \gamma_i) P_i = \sum_{i=0}^{L} \alpha_i P_i\right);$$

$$A_2 := \widehat{A}_2 + \sum_{i=1}^{L} p_i \widehat{A}_2 \quad \left(= \widehat{\beta} \cdot \widehat{v} \sum_{i=0}^{L} P_i = \widehat{\beta} V\right).$$

Then for all $i \in \mathcal{D}$ the values B_i are constructed as

$$B_i := (\widehat{r} + \gamma_i) V_i \quad \left(= (\widehat{\alpha} + \widehat{\beta} + \gamma_i) V_i = (\alpha_i + \widehat{\beta}) V_i\right).$$

Lastly, we set $r_0 = \widehat{r}$ and for all $i \in \{1, \ldots, L\}$

$$r_i = \widehat{r} + \gamma_i \quad \left(= \widehat{\alpha} + \widehat{\beta} + \gamma_i = \alpha_i + \widehat{\beta}\right).$$

We observe that the DSD adversary $\mathcal{A}$ can also compute B_i's herself, which is not surprising given the fact that she has the corruption power to know all secret x_i's (and c).

If $\widehat{r} = \widehat{\alpha} + \widehat{\beta}$ (and then in the DSD scheme for all indices $r_i = \alpha_i + \widehat{\beta}$), we are in the normal situation. However, if $\widehat{r}$ is random, then all the other response values are random as well. This construction yields a valid input to our DSD adversary, and can hence be used to break the privacy of the Randomized Schnorr scheme. □

5.5 Discussion

There is an essential difference when working with a simple pair of private and public keys, and with Attribute-Based Credentials. By definition a private key should never be revealed, whereas attributes, carrying valuable information about an entity, may sometimes be disclosed. The Randomized Schnorr scheme [BCI08] employed the former model. We intended to generalise the scheme and apply a similar technique to attributes. It is crucial to use a Pedersen commitment (see Definition 3.7 on page 47) which has a randomly chosen secret value that should never be revealed. This secret value (x_0) ensures that the commitment is never fully opened and thus, other entities cannot claim the same identifier. Technically, we required that $0 \notin \mathcal{D}$ and $0 \notin \mathcal{E}$.

The results of the work of Lee et al. [LBSV10] demonstrate that five point multiplications are feasible on a passive RFID tag. When a similar hardware architecture is applied for the DSD scheme, we foresee practical example applications with selective disclosure. Tags can have for instance two attributes (and an additional private scalar of course). If such a tag discloses 0, 1, or 2 attributes from the remaining set, the number of required point multiplications is 4, 5, or 6, respectively.

Also, favourable remarks are valid for the memory requirements of a single tag. Assuming L attributes, a tag has to store $L+1$ values where each is 160 bits long as the group keys in the hierarchical proof protocols in Batina el at. [BSSV12]. Having, for instance, four attributes to store, a tag requires 800 bits memory (assuming a curve over a 160-bit field). This is completely acceptable even for passive tags as attributes could be stored in the ROM memory the same way as the ECC parameters. (Unlike registers, ROM is considered to be cheap.)

RFID chips in (biometric) passports are real-life examples of a similar logical setup. Passport numbers, or identifiers, can be checked in an authentic database, while attributes can be locally queried. Access to biometric data requires an additional key[74]. The difference between the current practice and the technique described in this chapter is mainly privacy and security related. Selective disclosure of attributes and designated verifier proofs prevent illegitimate entities from retrieving personal information.

Another practical consideration is the governance of verifiers' public keys. There are two clear options in terms of public-key management when setting up a real-world system. Both of them guarantee security in different ways for a prover's identifier and disclosed attributes. First, an RFID tag may store all the public keys corresponding to the identifier (V) and all public attribute keys ($V_1, \ldots, V_L$). In this case at each protocol execution the prover discloses all requested attributes using those designated verifier keys. Second, a tag may use an instance-based designated verifier key. During interrogation a reader gives its public keys with a certificate along with its proof request.[75] A tag checks the certificate to disclose attributes with using only verified designated public keys.

We applied the idea of attribute-based authentication to the RFID technology. As a result powerful and flexible use-cases can be developed. Considering the example of hierarchical proofs [BSSV12], our solution could be deployed meeting exactly the same requirements as envisioned by the tree structure of the hierarchical proofs. To obtain the same functionality, one could sort the attributes according to their order of importance. More precisely, choose x_1 to be less important *i.e.* less privacy/security critical and therefore, the first secret verification key v_1 can be stored on a lot of readers, while v_3, for example, only at a very limited set of verifiers, etc. This infrastructure can easily be incorporated in the DSD scheme. In this way, we achieve not just a more fine-grained access control for tags, but also more fine-grained permissions for readers.

[74] The so-called Extended Access Control (EAC) includes this functionality.

[75] We assume here an underlying PKI for which the prover stores the root CA's public key.

Besides RFID applications, we foresee other possibilities. The idea of designated verifier can also be applied together with credential signatures[76] integrated in the U-Prove or Idemix technology. Issuing does not need to be changed, but verification is improved. The basic idea is as follows. During interaction with the verifier, the prover sends her credential signature, *i.e.* the issuer's signature over her identifier. An Idemix credential signature can be randomised every time, while the ones in U-Prove should not be reused.[77] As a result, a system can realise a full-fledged ABC system without the need of an underlying secure channel. This makes the designated verifier selective disclosure proof a powerful cryptographic technique.

[76] This is proposed in one of our student papers [PAL13].

[77] To achieve multi-show unlinkability each credential can only be used once as explained on page 58.

Chapter 6

An ABC Ecosystem

> "*[A]rchitecture is politics.* The structure of a network itself, more than the regulations which govern its use, significantly determines what people can and cannot do."
>
> Mitch Kapor, 2006

We have made a long journey. After seeing the main problems with current identity management practice, we studied Attribute-Based Credentials and their smart-card implementations. We also considered a more specific technical problem about how an ABC card and a verifier can communicate securely.

In this chapter we propose an infrastructure based on our group's smart-card ABC implementation [VA13, Vul14] for tackling several of the problems in IDM. The fundamental information unit about an individual is an attribute in an identity management system based on such an ABC card. This approach will have a lot of benefits, including easier authentication and more transparency for users, more reliable interaction between users and organisations, and more user-centric and more privacy-friendly identity management, in general. We will discuss possible approaches to an ABC ***ecosystem***, an independent infrastructure based on this technology. This chapter describes an ABC ecosystem in which attribute-based identity management is accomplished.

We intend to put Attribute-Based Credentials in practice because we believe that this technology should be widely deployed. Its flexibility and privacy properties are exceptional and make it suitable for many applications. The examples include a national identity system [BKPR14] that can provide a reliable source of authentic information within and beyond government administration. By making ABCs practical, we are able to build a user-centric and privacy-friendly identity-management system realising the objectives described in *e.g.* [CSFH$^+$05, BSCGS07] within the European legal framework, see the EU Directives 95/46/EC (Data Protection, [Eur95]), 2002/58/EC (Privacy and Electronic Communications) and the proposal of the General Data Protection Regulation (*cf.* page 19). Since the mapping of personal data to separate attributes is easy to comprehend, it is also easy to argue about attribute-based, privacy-friendly access to services. Our hope, supported by some positive experience, is that people are increasingly willing to accept that it is sufficient to authenticate based on those, possibly non-identifying attributes. In sum, although ABCs are devised primarily for privacy-friendly authorisation, we intend to show that with an ABC card an identity ecosystem can be built that realises much more.

Our goal in the chapter is to summarise our vision about the possible usage scenarios and an infrastructure based on ABC cards.

6.1 Introduction

Authorisation requires authentication: Before letting someone do or use something, it must be clear that this someone is actually allowed to do so. Traditionally, authentication is understood as a proof of identity, for instance, by means of a password or an identity document. But precisely identifying people, using uniquely identifying numbers and names – such as a social security number (SSN), credit card or bank account number – is often an overkill. In many situations it suffices to know some attributes (properties) of a person in order to authorise a transaction. If a hairdresser offers a cheap haircut to students, it is not necessary, or even desirable, that the hairdresser learns, and potentially stores, a (uniquely identifying) student number as part of the proof of "studentship". Similarly, buying alcoholic drink only requires a proof that the buyer is above a certain age limit (16, 18 or 21). Attribute-based authentication aims to provide a mechanism for exactly doing this: Allowing transactions on the basis of those attributes which are required for the transaction. The main advantages are:

- It is privacy friendly: in the sense that it is based on the idea of data minimisation and that it provides unlinkability among user transactions;
- It offers protection against identity fraud: if one's identity is not involved in a transaction, it cannot be stolen;
- It provides a new, more flexible approach in identity management and authentication: in particular, an approach that is based on attributes instead of whole identities with unique identifiers.

Attribute-based authentication is not new. Attribute certificates [FH02] were defined in the X.509 stack over a decade ago. They enable authentication that does not require identification; *e.g.* role-based access, proof of membership or more generally attribute-based access control [WWJ04, YT05, VFK$^+$14]. However, they are (1) linkable (each transaction is linked to the same public key) and (2) transferable (delegateable). Attributes in the context of Attribute-Based Credentials and in this chapter are different; they provide security, unlinkability and untransferability, simultaneously. Cryptographic techniques that enable secure and privacy-friendly attribute-based authentication have also been around for more than a decade, see [Bra00, Ver01, CL01, **?**]. But what is new is that the state-of-the-art smart cards are powerful enough to perform the required (non-trivial) cryptographic operations in an adequately efficient manner. Hence only now we see efforts to actually deploy attributes in practice. This chapter is based on the experiences in one such deployment in the course of the IRMA project[78] in The Netherlands. It relies on the Idemix technology [**?**, CVH02, IBM12] and uses personal smart cards as carriers of credentials and attributes. Getting privacy-friendly attribute technology up and running brings us into largely unexplored territory that poses a multitude of technical, organisational and research challenges. As its main contribution, the current chapter explores these matters. It concentrates on the issues that arise regarding the organisation of multiple attributes and of the dependencies among them, and on the decisions that need to be made to make these cryptographic techniques and their implementation practical while preserving their advanced properties. We refer the reader to Chapter 3 for the underlying cryptography and the main results of the applied smart-card implementations.

To the best of our knowledge, there are only two other pilot projects – one in Sweden, the other one in Greece – in the context of Attribute-Based Credentials. Both of them are carried out by the EU-sponsored ABC4Trust [CKL$^+$11]. The Swedish pilot [BGOZ12] gives anonymous access for elementary school pupils to online resources (*e.g.* chat rooms), while the Greek pilot [ALP$^+$12] enables university students to evaluate lectures anonymously. In both cases eligibility and privacy are of primary importance. Although our pilot uses the same underlying technology, the objective of our research is more general as we investigate a *broad variety* of attributes and applications. The kind of challenges investigated in this chapter do not appear in these ABC4Trust pilots since each focusses on a single context.

In Chapter 2 identity was defined as the set of all characteristics that have been attributed to this entity within a scope. So, from the scope's perspective the entity is seen as all his or her characteristics. From the entity's point of view, however, each of these collections is a ***partial identity*** [BMH05, PH10]; this is best depicted by the FIDIS project – see Figure 6.1. Although authorisation often requires an identified partial identity,[79] it would be sufficient to extract directly those attributes that are relevant from a partial identity in a specific authorisation decision. This would make the process *more efficient* with the participation of potentially *fewer parties* (no IdP online), and most importantly, it would often lead to a *more privacy-friendly* transaction because the required attributes can truly be the minimal data

[78] I Reveal My Attributes (IRMA); see `https://www.irmacard.org`. This abbreviation and its full form were coined by Bart Jacobs and Pim Vullers, respectively, during a brainstorming session about the pilot project at the lunch table in the courtyard of our university's Huygens building. In May 2012, the project name was approved by our group's secretary Irma Haerkens who has the same first name – not coincidentally.

[79] In most cases in practice, a typical authorisation process scheme looks like (*cf.* network-based IDM in Figure 2.1): (U) Claim an identifier ⟶ (U) Prove that this identifier belongs to you ⟶ (IdP) Look up the partial identity corresponding to this identifier ⟶ (IdP) Select relevant attributes from this partial identity ⟶ (IdP/SP) Make an authorisation decision based on these attributes (possibly with additional data).

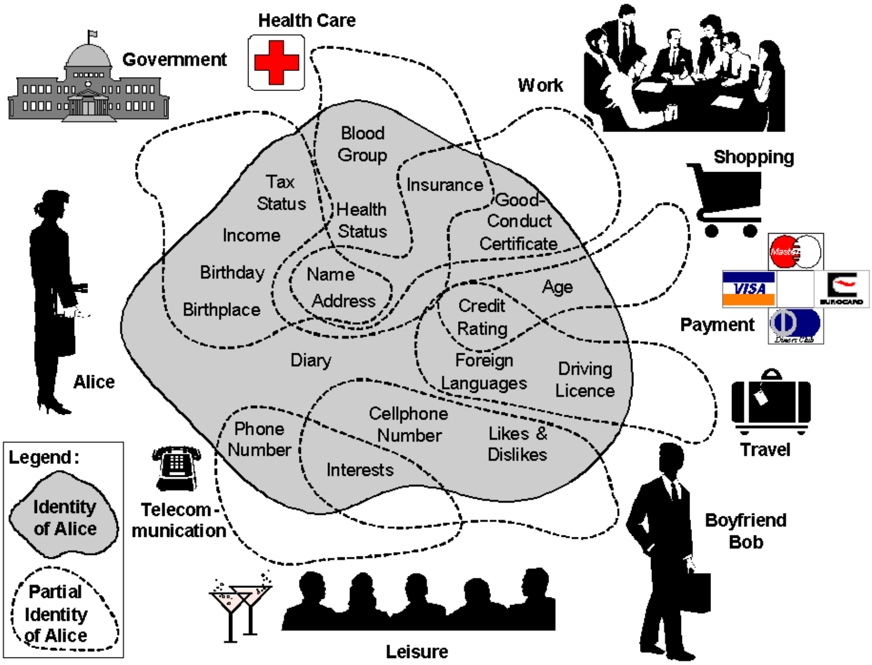


Figure 6.1: An individual's identity is made up of attributes which form several partial identities in different contexts. (Source: FIDIS [Bac05, Figure 7])

set required for the access decision. So, instead of all attributes from a partial identity, only a minimum subset is used.

We can imagine that with a personal ABC card, people manage dozens of attributes that are relevant for authorisation purposes in all kinds of contexts. Given that there are many dependencies between all these attributes, the question of how to organise them in a logical, coherent and intuitive manner is non-trivial and often not free from politics (*e.g.* potentially enforced Facebook's real-name policy; see Section 6.4.1). In this chapter we make the various issues explicit that we came across in the context of the IRMA project and explain the choices we have made. This is certainly relevant beyond this particular project as well as this thesis.

As we mentioned our discussion addresses smart cards as ABC carrier devices for two reasons. Firstly, current smart cards are able to perform all cryptographic operations required for achieving the favourable properties of ABCs. Secondly, individuals have a familiar and intimate relation with their smart cards. They recognise them (particular cards are easily distinguishable based on their outward appearances), know the technology (bank cards, public transport cards, *etc.*) and can trust them more than other devices (such as their mobile phones susceptible to malware).

The chapter proceeds with a collection of possible Attribute-Based Credentials and related use cases, then we give a possible description of *attribute-based identity management* (ABIdM) in Section 6.3. Sections 6.4 and 6.5 study the new notions of *credential design* and a *scheme manager* in the context of ABIdM. Section 6.6 briefly discusses revocation and the use of a personal mobile phone to communicate with an ABC card. Finally, we consider other attribute-based technologies in Section 6.7.

6.2 Motivation: Use Cases

To motivate attribute-based identity management, we describe some envisioned credentials and use cases. Informally, we assume that issuing and showing protocols work as they are described in Chapter 3 and that an Attribute-Based Credential can hold at most four attributes. As the current discussion considers attributes of a wide variety, we let attributes be non-identifying as well as identifying. While ABCs were originally devised for anonymous applications, we are convinced that they provide many more usage and application opportunities with (partly) identifying attributes.[80] These use cases serve as motivation for credential design (see Section 6.4) principles and more general concepts relating to an ABC ecosystem.

Figure 3.1 gave an abstract view of an Attribute-Based Credential. In this section we discuss example credentials with a main focus on the attributes within credentials. We describe possible issuers at each credential. The template of the simplified abstract view in this chapter looks like this:

credential name
attribute 1
attribute 2
attribute 3
attribute 4

Attributes can be of different types: boolean (‘`yes`’ or ‘`no`’), an option from a list (*e.g.* ‘`gender`’) or a string (*e.g.* ‘`full name`’). Mostly attribute types are clear from the description and the context.

Age bounds The attribute that is most needed in practice is probably the minimal-age attribute, like “over 18”. It is useful for many online and offline transactions, such as buying (violent) games, alcoholic drinks, cigarettes, (certain) movies or books and online gambling. Analogously, one may form maximal-age attributes, like “under 15”. They may be used to regulate access to certain chat rooms which are set up exclusively for minors.

[80] In this thesis the degree of anonymity is out of scope, *e.g.* k-anonymity [Swe02, MKGV07], although we recognise that it is an essential consideration in the context of attributes and privacy.

Within the Idemix context there are “interval proofs” which make it possible to derive these minimal- and maximal-age attributes from the date of birth. Such proofs are computationally rather expensive and are (currently) not included in this project because they cannot be implemented on a smart card.[81] Instead, minimal-age and maximal-age credentials are foreseen that contain the most often used age limits (of boolean type). These credentials look as follows:

minimal junior
≥ 12
≥ 16
≥ 18
≥ 21

minimal senior
≥ 60
≥ 65
≥ 70
≥ 75

maximal junior
< 12
< 16
< 18
< 21

The most authoritative issuers for such credentials are local or national authorities, using, for instance, their citizen registration database.

Citizen Identity A citizen’s identity may be organised in three coherent credentials:

name
full name
birth name
family name
first name

identity
social security nr.
date of birth
place of birth
gender

address
country
city
street + number
postal code

As before, governmental authorities are the most authoritative source to issue such credentials. Recall that each of these attributes can be used separately in authentication. But also combinations of these (and other) attributes are possible.

Loyalty Cards and Pseudonyms Shops, airlines and other commercial organisations like to build a steady relationship with their customers using loyalty cards. A possible and popular means is to give them selected benefits when they have accumulated enough loyalty points. Applying such cards, these companies can keep track of who purchases what, and this allows them to build up detailed profiles of their customers. In practice, each chain of shops issues its own (physical or virtual) loyalty card. This is no longer needed with an open ABC card, since each organisation can add its own loyalty credential to it which can work independently.

[81] After this chapter was written new implementation results showed that with certain cryptographic modifications and some efficiency loss, equality and interval proofs are becoming feasible on current smart cards. [dlPHV14]

shop *X* loyalty
customer number
customer status
...
...

The customer number in the credential acts as a key for a database entry in the back office that contains the actual purchase history of the customer (card holder).[82] On the basis of this history, a customer may reach a certain status, like bronze/silver/gold. In each shopping situation the customer may be offered the option (i) to buy anonymously, using only the status attribute to get certain benefits, or (ii) to buy non-anonymously using also the customer number. Only in the latter case, the purchase is added to the personal history (in the back office) and contributes to the status build-up. The remaining two attributes, written as '...', are left open and can be used for other Customer Relationship Management (CRM) purposes. They can also be left empty (blank).

A card holder may use his card with this credential offline, in a "brick and mortar" shop. But it can also be used online, to purchase something, or to access an overview of the card holder's purchase history and, possibly, to update the status attributes. For these purposes, the loyalty number attribute is sufficient as authentication.[83] Of course the name and the gender are appealing to have for communication purposes, but they are not necessary to be stored in this credential or in the CRM system. Moreover, the customer may choose not to use his real or authentic data. An address credential may be required in case of delivery. It can be verified in each transaction, and need not be stored centrally.

Such customer numbers in credentials may thus be used as pseudonyms, one for each commercial relationship (with shops X, Y, Z, *etc.*). There is a potential privacy risk when many commercial organisations decide to cooperate and use one number for all of them. In this way they can profile customers across different organisations, a bit like it is done now using third-party cookies or device fingerprinting [MM12]. Such broad commercial use of a single pseudonym, possibly at a national level, may be forbidden by the scheme manager (see Section 6.5) and/or by the relevant data protection authority.

Medical information In a medical context one can envisage attributes for patients and for medical staff. Patients can carry for instance credentials with attributes containing essential personal medical information in a micro-dossier, see the first two credentials below. (Lists, like (encoded) allergies and chronic diseases, are assumed to be disclosed as a whole.) Medical staff can use credentials that de-

[82] A similar approach for a privacy-friendly loyalty system was recently proposed in [MDPDD14].

[83] Technically, a secure channel is established between the card and the verifier in which the card also provides an anonymous validity proof as described in Chapter 4.

scribe their medical role and access rights to patient files, as suggested in the third one:

medical basics
blood type
allergies
chronic diseases
…

medicines
…
…
…
…

medical staff
position
registration nr.
…
…

The first two credentials may be issued by health authorities (hospitals, or even general practitioners). They are useful in medical emergency situations, like after an accident. The last credential falls under the responsibility of health employee registration authorities. The 'position' attribute typically determines access right to medical records, such as: doctors may both read and write, but nurses may only read. For accountability, the registration number should be used in each such transaction in order to monitor who accesses which file.

As we see, further personal details, such as name or date of birth, are not included in these credentials. The citizen identity credentials already describe those issued by the governmental administration. Those attributes may also be revealed in a medical context. Attribute duplication is not necessary and not even desired. We elaborate on this issue later; see Section 6.4.

Access control and role/attribute-based access control Within an organisation X or an educational institution, a credential can be designed for specific access rights, roles, positions, *etc.*, as suggested in:

***X* access**
parking
main entrance
vault
intranet

***X* staff**
position
employee nr.
…
…

student
university/college
field of study
student nr.
enrollment year

Issuing a mobile phone number credential So far we have concentrated mostly on the attributes in credentials and how they are verified. We now consider the issuing of a credential. Suppose one wishes to obtain a credential containing ones mobile phone number. The obvious issuer is the mobile network operator (MNO). The issuing procedure might work as follows:

1. The user goes to the website of the MNO, using TLS (*i.e.* https), and proves her name and date of birth using her ABC card.

2. The MNO looks up in its database if there is a contract with this name and date of birth[84]; if not, it aborts; otherwise, it sends a one-time code over SMS to the (mobile) phone number associated with this contract.
3. Upon receiving this one-time code, the user feeds it back into the website (within the same https session).
4. The MNO now issues the credential containing her phone number, possibly together with some other attributes, to the user's card.

What is interesting about this protocol is that it involves authentication that uses both existing credentials and an out-of-band channel. The use of existing credentials leads to dependencies among credentials, as described in Section 6.4.1.

Login Currently, users have to remember several online usernames and corresponding passwords at various websites. An alternative way is to have the website issue a credential when the user registers; such a credential would contain the '`username`' attribute. The login procedure is simply showing the username from the card – the password is not required anymore.

Festival ticket We conclude this list of use cases with a non-standard application of attributes, in order to suggest the great variety and breadth of possible usage scenarios. If individuals wish to get a ticket online for a pop concert or other festival, they currently need to fill out long forms requiring personal information. The main purpose – apart from allegedly profiling – seems to be to prevent copying or transfer of tickets and to limit the number of tickets one individual purchases. Physical (paper) tickets are traditionally anonymous. One may also provide such a ticket in electronic form, after payment, as a credential for the festival at hand, containing for instance the following attributes:

festival ticket
festival name
date
ticket number
...

where ... may describe any additional information, such as pre-paid consumptions. The issuing of such a digital ticket may be preceded with some verification of attributes on the same card. For instance, it is required to check that the card

[84] In many countries, before obtaining a mobile phone subscription, a contractual commitment is required which includes the verification of an identity document and the storage of some personal data; this is assumed here.

holder is ‘over 18’ to receive a voucher for alcoholic drinks. Upon entering the festival terrain, the presence of a valid ticket on a card can be checked (and consumption vouchers can be handed over). The next day the ticket/credential is unusable, and can be removed from the card (by the card holder).

6.3 Attribute-Based Identity Management

Identity management was defined in Definition 2.2 as “the processes and all underlying technologies for the creation, management, and usage of digital identities”. A user-centric identity management system “needs to support user control and consider user-centric architectural and usability aspects” [BSCGS07]. More generally, user-centricity refers to the quality of a system that it is “structured so as to allow users to conceptualise, enumerate and control their relationships with other parties, including the flow of information” [CPR09].

Our goal in this section is to embed the ABC technology and its smart-card implementation into a user-centric identity management framework, which results in what we call attribute-based identity management (ABIdM). To this end we discuss the underlying technologies and processes. Our narrative here is organised from the bottom up, that is, from the ABC card through the participants and the assumptions about the technology to the processes in the identity ecosystem. The section concludes with some practical considerations that we encountered during the IRMA project.

6.3.1 Participants and Assumptions

In an ABC ecosystem we distinguish at least four types of operational participants: the user, the card provider, the issuer and the verifier. Each ***user*** (an individual) has a personal ABC card that carries credentials with attributes related to the user. Such a card is provided by the ***card provider*** to a user after a card provisioning process. The card is then bound to the user (card holder) physically as well as digitally. An ***issuer*** issues certain attributes to the user’s card in the form of a credential after verifying those attributes with respect to the user. A ***verifier***, protecting some resource (*e.g.* , application, service), checks attributes from the credentials on the user’s card and decides upon user access based on the attribute values. Later we mention further possible participants, such as a scheme manager and a certificate authority (Section 6.5) or a revocation authority (Section 6.6.1).

All underlying cryptographic techniques are assumed to be secure with regard to the system’s security parameter. In particular, ABCs have to provide all security and privacy properties described in Section 3.3 on page 52. All ABCs on a user’s card are bound to the card by a ***secret key***, no credential can thus be transferred to another user (provided that the card itself with its Personal Identification Number (PIN) is not transferred). This secret key is generated on the card, embedded as an attribute in each credential and is never revealed. A card is assumed to be tamper-resistant so that it protects all secret values (keys, PINs) and attributes. It has to be able to perform all required computational tasks. It is thus assumed to generate all random values unpredictably, carry out cryptographic computations correctly. The card implementation is presumed to execute also conventional cryptographic operations for the whole protocol run. For instance, signature verification is required for checking the public-key certificate of a terminal (of an issuer or a verifier), and encryption is necessary for communicating with the terminal through a secure channel. The entire software implementation is supposed to be side-channel resistant to prevent information leakage about any secret values.

The communication between a card and a terminal of an issuer or a verifier is assumed to be protected by a secure channel defined and realised in Chapter 4.[85] In offline use cases this secure channel provides anonymity leaving no linkable traces of a card. However, in online use cases it is assumed that the underlying transportation of messages is anonymous, such as a mix network or Tor[86] in particular. Otherwise, the Internet connection of the user's card-reading device (*e.g.* PC or mobile phone) makes the communication traceable. To prevent rogue issuers and verifiers, conventional public-key certificates are applied. We discussed that an Attribute-Based Credential is a special signature on the attributes; the issuer's corresponding public key has to be certified. In this way a verifier can ensure during a verification protocol instance that an ABC's signature originates from an authentic issuer (certificates are assumed to be publicly available). Verifiers also have to have certificates for the cards to be able to justify authorisation requests. Finally, cards are assumed to provide random identifiers (specified in ISO14443-3 [ISO11]) every time when they communicate with verifiers; otherwise, card transactions would be linkable, based on this underlying number.

6.3.2 Processes

Now we turn to the operational processes in an ABC ecosystem from the user's point of view. The processes below show a possible realisation of an identity system based on ABC cards. Several concepts will be discussed in more detail later.

- Card provisioning phase
  - Card registration. This is a user-initiated, offline or online process in which a user intends to receive a new card. This step includes the registration of fundamental personal data; in particular, an official (self-asserted) photo, an identifier in the card provider's system (such as a random value, a pair of a name and a date of birth or a social security number) and an address (physical or electronic).

[85] Alternatively, the system can be set up in a way that ABC proofs also support designated verification (see Chapter 5).

[86] `https://www.torproject.org` [last accessed: October 26, 2014].

  - Card personalisation. The smart card is prepared for the user. This step includes printing on the surface of the card and installing the required software components on its chip. The user receives a notification to her provided address that includes a temporary operational PIN.
  - ID Proof. The card provider authenticates future card holder. This is a physical process in which the prospective card holder has to be personally present at the organisation and to prove her identity using her photo and prescribed identity documents (*e.g.*, passport or driving licence) belonging to her.
  - Vouching for identity. The first (root) credential is issued to the card. In this process the user also has to enter the operational PIN received to her address. With this credential the user can prove her identity later (see details below and in Section 6.4).
  - Card hand-over. If the entire procedure above succeeds, the card carries all necessary personal information (photo, root credentials, *etc.*). The card is bound to its holder both physically (photo) and digitally (PIN), and so it can be handed over to the holder.

- Verification (*i.e.* authorisation)
  - Access request. The user initiates communication with a verifier by requesting access to some resource. The verifier informs the user which attributes need to be checked. The user gives her consent by presenting her card to the system. Technically, the verifier's certificate proves that the verifier legitimately requests these attributes. The card can verify the certificate and – if it is valid and the user has given her consent – reveal the requested attributes.
  - Proof of card possession. The user authenticates to the card[87] to guarantee that she is the legitimate card holder. In offline scenarios the photo may be sufficient, in online scenarios the PIN is required.
  - Disclosure of attributes. The card and the verifier perform an interactive ABC selective disclosure proof. As a result, the verifier receives authentic attributes about the user.
  - Access decision and access. The verifier makes a decision based on the received attributes and grants or denies access to the resource.

- Credential issuance
  - Credential request. The user requests a new credential onto her ABC card.

[87] The card is trusted by the user and all other parties, so this authentication guarantees privacy for the user while provides assurance for issuers and verifiers. Colloquially, we can say that the IdP (issuer) delegates its job to the card; *cf.* footnote 79.

  - Proof of card possession. See within the verification procedure above.
  - Authorisation. The issuer verifies eligibility of the card holder for the new credential by possibly using attributes from the card. (See a further discussion about credential types based on authorisation on page 120.)
  - Issuance. After authorisation, the issuer collects the necessary attributes and issues a new credential to the user's card according to the ABC protocol.

Clearly, the processes support a user-centric design for identity management by "giving the user full control of transactions involving her identity data" [BSCGS07]. Card provisioning is initiated by individuals. Therefore, users themselves decide to participate in an ecosystem. Issuing procedures are initiated also by users whenever they need new or updated credentials. Finally, verification is initiated and consented by the user.

6.3.3 Card and its owner – practical considerations

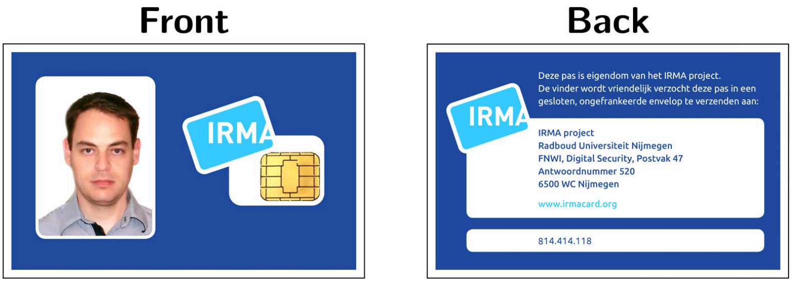
Since ABC cards provide a high degree of privacy and they work dynamically, several new practical decisions have to be made when putting forth general recommendations about an ABC ecosystem. In this section we explore three aspects of the relation between the card and its holder: PINs, the outside of a card and the card management software interface.

Only the card holder should be able to use her ABC card. In general, a PIN can restrict access to a card by ensuring:

- *confidentiality*: to prevent unauthorised reading of private data, for instance, after a card loss;
- *user consent*: to make sure that a card is only used when the card holder agrees;
- *authentication*: the card is only usable by the card owner; in particular, someone else who obtains/finds a card cannot use it.

In general, the addition of new credentials to a card should be protected by a PIN to guarantee consent and authentication. But when should revealing of attributes be protected by a PIN? Although the "over 18" attribute seems fairly innocuous, it should not be possible that a minor temporarily borrows an adult's card to obtain "over 18" items online. Hence, the age credential should be PIN-protected. Attributes that give access to a parking or open an entrance may typically be not PIN-protected, except for high-security facilities.

If some attributes or credentials require PIN-protection and others do not, the question arises: Who decides about this? One option is to leave the decision to set PIN-protection to the verifier's card-reader terminal or the user. Another one, is to follow some general policy, which seems to be more practical in terms of privacy,



Deze pas is eigendom van het IRMA project.
De vinder wordt vriendelijk verzocht deze pas in een gesloten, ongefrankeerde envelop te verzenden aan:

IRMA project
Radboud Universiteit Nijmegen
FNWI, Digital Security, Postvak 47
Antwoordnummer 520
6500 WC Nijmegen

www.irmacard.org

814.414.118

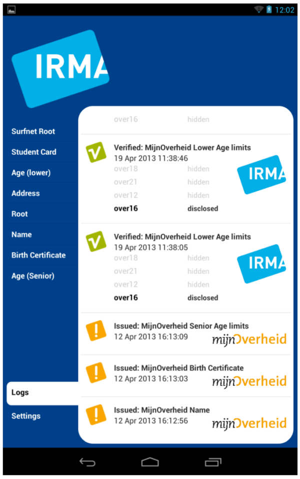
Figure 6.2: An example ABC card (within the IRMA project) showing minimal personal information.

security and usability. This policy should be set in general terms by the scheme manager (see Section 6.5), and elaborated in detail with each credential issuer. For instance, such a policy can be that 'the use of a PIN is mandatory online and it is described in the verifier's certificate offline'.

An ABC card is primarily used as an enabler for privacy-friendly authorisation online as well as offline. In online usage the outside of a card is irrelevant for the issuer or the verifier. The only practical requirement is that the card owner should recognise his own card (to prevent confusion). In offline scenarios, however, the (human) verifier should be able to check that the person presenting a card is the card holder. This is done using the holder's photo on the front of the card. Besides this picture, the verification of certain attributes may be strengthened by requiring the PIN to further improve security.

On the back of a card there is (general) information about how lost cards can be returned. Additionally, there is a card-specific number. It can be used to search for the card during card hand-over and to look up the owner of a lost-and-returned card. The card number is a dangerous addition that could make it possible to trace cards. Therefore, the card number is used only externally, and not internally, in the chip. Figure 6.2 shows an example for such an ABC card.

Unlike smart cards used nowadays, the digital content of an ABC card can be dynamic. Today I may only have an identity credential, but after issuing new credentials tomorrow, I can have several credentials and dozens of attributes on my card. Since smart cards do not have a user interface, it is desirable to provide functionality for users to see and manage their cards' content. This is the goal of the ***card management*** application which can be run on a PC, on a mobile phone or on another device that has a (contact or contactless) card reader. This novel application enables the user to administer her card privately. The application should run on a trusted device (trusted to display information honestly and not to reveal/forward personal data to any third party) and be protected with a separate (potentially longer administration) PIN.



over16 hidden

Verified: MijnOverheid Lower Age limits
19 Apr 2013 11:38:46
over18 hidden
over21 hidden
over12 hidden
over16 disclosed

Verified: MijnOverheid Lower Age limits
19 Apr 2013 11:38:05
over18 hidden
over21 hidden
over12 hidden
over16 disclosed

Issued: MijnOverheid Senior Age limits
12 Apr 2013 16:13:09

Issued: MijnOverheid Birth Certificate
12 Apr 2013 16:13:03

Issued: MijnOverheid Name
12 Apr 2013 16:12:56

Figure 6.3: An example interface for the card management application in the IRMA project. It shows the last few activities carried out by the card.

- Credential administration. A user can maintain all credentials on her card: overview of existing credentials, initiate credential update, delete expired credentials.
- Logs. A user can look at the last few events on her card and check that the card was not abused. Such recorded events are the issuance of a credential and each verification instance (with the identity of the verifier and the revealed attributes); see Figure 6.3.
- Modifying PINs. The user can change both (operational and administration) PINs. As the first use of the card management application, the holder should change her system-generated PINs right after card provisioning.

6.4 Credential Design

An ABC card may contain over a dozen credentials, each with multiple attributes. In a particular selective disclosure proof, any subset of attributes in a single credential may be revealed, without revealing the remaining attributes.

Within the context of the IRMA project at most four attributes are grouped together in a credential. The number four is chosen pragmatically, mainly for implementation reasons, but other reasons turn out to confirm this choice. On the one hand, having many attributes in one credential means that if only one attribute is revealed, all the others remain hidden. Hiding more attributes requires more time, and thus reduces the performance (see Section 3.4, page 59). On the other hand, the number four seems to be reasonable to form a coherent set of attributes, issued jointly by a single authority.

All credentials are required to contain two fundamental attributes. First, each user has a *master secret key*, stored in the smart card's secure storage, which is also incorporated – technically, like an attribute – in all credentials. Second, an *expiry date* has to be determined at issuance, and it is included as an attribute applying to the whole credential. When the credential is verified, the expiry date can be revealed to confirm validity.[88]

In the credential examples on page 111 we have seen that a card holder's name occurs in the Name credential (obviously!), but not in a medical staff or employee credential. This may look strange at first. In principle, there could be multiple name attributes, issued by different parties (like local authorities and Facebook). Similarly, multiple accounts at different banks or different phone numbers can be issued in separate credentials. It is the role of the scheme manager to decide which organisations are authoritative about a type of credential. Verifiers can then decide which issuer they wish to trust for having attested to certain attributes. However, we propose as few attributes to be issued by multiple issuers as possible for simplicity and efficiency. Private data should be kept accurate and up-to-date according to the data quality principle [OEC80]. First, the quality is guaranteed most likely by the most authoritative entity. Second, it is not straightforward to keep the same piece of data up-to-date at different locations. In fact, so far we are excluding any duplication of attributes (same content, different issuers). Although theoretically feasible, we think such attribute duplication makes credential administration much more confusing for card owners (and possibly for verifiers too). This means that for each possible attribute there must be a single most authoritative issuer. Again, it is the role of the scheme manager to decide who is authoritative about which set of attributes.

6.4.1 An Example Credential Dependency Graph

As we saw in Figure 3.1, credentials are containers of attributes signed by an authoritative issuer. An issuing procedure requires some form of authentication to prove that a specific card is entitled to hold a credential. This authentication can include the verification of already existing credentials on the card. On the one hand, a so-called ***root credential*** does not require the verification of other credentials from the ABC card before it is issued. This process can be performed during card provisioning or it can be initiated later by the user. In particular, root credentials are required when a user establishes a new identity, *i.e.* in a new context. Often this process requires an out-of-band authentication (*e.g.* a citizen root credential requires a physical verification of the user), while in other cases the process does not require authentication (*e.g.* an online commercial service may not require verification in order to issue a pseudonymous loyalty number). A ***dependent credential***, on the

[88] For privacy reasons, the expiry date should be a coarse date, such as the last day of a month.

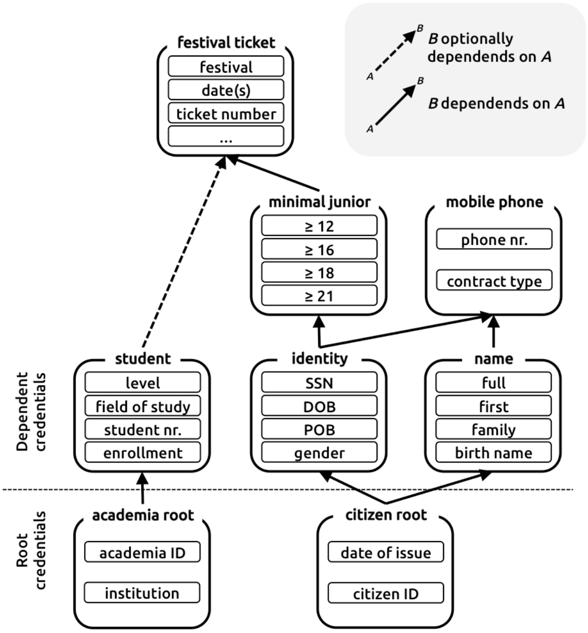


Figure 6.4: An example for credential dependencies on an ABC card.

other hand, is issued only after verifying at least one other existing credential on the ABC card. First a set of attributes (or possibly, the presence of a credential by using an empty proof) on the card is verified, and then the new, dependent credential is issued on the same card. (We note that the two protocols have to be performed in the same session, *i.e.* in the same secure channel.) For instance, the issuing procedure of a 'mobile phone number' credential (described on page 112) results in a typical dependent credential because the user's '`name`' and '`date of birth`' are verified from other credentials residing already on the card.

All credentials on a card can be depicted using these dependencies. Figure 6.4 shows an example for a simple dependency graph. In this example, there are two root credentials. An 'academia' credential represents the card holder's identity in the national education system. A 'student' credential, for instance, relies only on the 'academia' root. After a student proves that he or she has such a root credential with the appropriate attributes of '`institution`' and a unique '`academia ID`', the organisation can look up all relevant personal data in its database and issue the 'student' credential. Note that this issuing procedure requires identification since a 'student' credential is bound to a specific person. Such a credential contains similar personal information as a traditional student card.

A 'citizen' root credential can be issued after a personal, face-to-face identification accompanied by a physical identity document authentication. This credential, similarly to the 'academia' root, contains an identifier in the given identity context; this may or may not be identical to the social security number [OHF07]. This root credential, issued by and often used within the government administration, can also be utilised in a broader context. For instance, a 'citizen identifier' can be used to retrieve attributes for an identity credential at the municipality, like those of an identity card. A trusted service (such as the municipality itself or another provider) can issue a 'minimal junior' credential after verifying the '`date of birth (DOB)`' attribute from the 'identity' credential. Strictly speaking, this procedure is non-identifying if we stick to the data minimisation principle. In the example on page 113, a 'festival ticket' is issued as a dependent credential relying on one or two credentials. A possible restriction is that such tickets can only be sold to customers '`over 12`', but in case a voucher for alcoholic consumption is also included, the ticket issuer must check the '`over 18`' attribute from the 'minimal junior' credential. Optionally, if the customer can prove that he is a bachelor or master student (using the '`level`' attribute from his 'student' credential), he can also get some discount. Note that all verification (authentication and authorisation) and ticket issuing can be carried out anonymously with the card's participation.

We discussed the possibility of logging in with a credential, previously issued by the website. In what follows, we take Facebook as an example in considerations that apply to many other similar organisations.

Facebook – root credential or not? If a user signs up for Facebook, she chooses the name that she likes (within certain technical/decency limits). Facebook has a Real Name Policy[89], but this policy is not enforced by some technical mechanism. Many people like to use a pseudonym on Facebook and currently this is possible.

Now suppose Facebook wishes to join the project at hand and would use ABC cards for authentication. The credential only needs to contain Facebook's user ID. An interesting question is: Should this be a root credential or not? This technical question has wide societal relevance.

1. People who do not wish to use their real name on Facebook expect Facebook's credential to be root, not depending on any other ones.

2. Facebook, on the other hand, probably does not want to have a root credential: It would like to first verify the (real) name on the ABC card (and probably more attributes), before issuing its own credential. In this way Facebook can enforce its Real Name Policy.

[89] `http://www.facebook.com/help/292517374180078`: "We require everyone to provide their real names [...] The name you use should be your real name as it would be listed on your credit card, student ID, *etc.*" [last accessed: May 5, 2014]. Later Facebook has changed its policy as follows: "We require people to provide the name they use in real life [...] The name you use should be your authentic identity; as your friends call you in real life and as our acceptable identification forms would show", referring to the options to "confirm your identity"; see `https://www.facebook.com/help/159096464162185` [last accessed: November 7, 2014]

3. Currently, it is possible to register with another person's name. Thus, to prevent this type of identity theft, individuals may also be interested to enforce the Real Name Policy.

6.4.2 Credential Design Principles

To summarise, we formulate five credential design principles that are meant as a set of non-mandatory guidelines to help a scheme manager (see the next section) or a community (see Section 6.5.4) to efficiently apply ABC cards.

1. Attributes in a credential form a coherent set.
2. All attributes in a credential fall under the responsibility of a single most authoritative issuer.
3. Attribute duplication (same content, multiple issuers) is avoided.
4. Verifiers can read only a limited, predefined, proportionate (in terms of the given context) set of attributes.
5. For efficiency reasons, credential design requires minimising the average number of credentials in verification processes.

6.5 Scheme Manager as the Defining Party

In order for an ecosystem to operate appropriately, a so-called scheme manager takes care of governance. This involves regulations in a contractual and a technical manner. The scheme manager determines possibilities and operations in an ecosystem although it does not take part in any of the operational steps. This section provides an overview of the main functions of this governing entity, discusses its power and proposes two different approaches to ecosystems in relation to the scheme manager.

6.5.1 Tasks

A scheme manager has a pivotal role in the operation and the possible success of an ecosystem. The following list collects the main tasks of a scheme manager:

- Initiation and implementation phase:
  - To define the main goal(s) of the ecosystem. (For instance, it serves as a national eID system or a loyalty card system.)
  - To decide about the technology (credential carrier device (*e.g.* the type of smart cards), ABC technology, cryptographic primitives (hash, signature, encryption mechanisms), *etc.*

  - To determine the main types of use cases and to lay down main regulatory and organisational principles, for instance by publishing an 'ecosystem charter'.

- Maintenance phase:
  - To design the card provisioning process and select a card provider.
  - To design (or to accept the design of) new credentials, issuing processes (*e.g.* , credential dependencies, identity proofing, *etc.*) and match them to potential use cases.
  - To certify issuers, which includes permission provisioning for the issuer which credentials it is entitled to issue to cards.
  - To certify verifiers (service providers), which includes permission provisioning for the verifier which attributes it is entitled to request from cards.

An informal description of certification processes (realising the last three steps above) can look as follows:

- Issuer certificate.
  - The issuer sends a request to the scheme manager for a credential design (see Section 6.4) describing the attributes that the credential would contain, the reason why this particular issuer is the most authoritative with regards to these attributes, the verification procedure that would take place before issuance and arguments why the credential design is in line with the general principles of the ABC ecosystem.
  - The scheme manager evaluates the request.
  - If the request is approved, the issuer generates a fresh key[90] for credential signatures that the scheme manager (or a delegated CA) certifies together with a description of the credential design.

- Verifier certificate.
  - The verifier requests attribute access rights from the scheme manager in relation to the authorisation type to a certain use case (*e.g.* to allow a user to watch movies online if she is over one of the age limits [12, 16, 18] and she is a member of the service provider's movie club). The request should include particular attributes within credentials that the verifier needs for this authorisation type and the verifier's public key that he would use in this application.

[90] Alternatively, the issuer re-uses an existing key which becomes certified also for this new credential.

  - The scheme manager evaluates the request whether it is in line with the ecosystem charter and it is well-supported (*e.g.* proportionate) in the context of this application.
  - Upon approval, the scheme manager (or a delegated CA) issues a new certificate containing the verifier's public key, the identifier of the verifier, a name of the application and the access right (or `Att_Acc`) that describes what attributes the verifier is allowed to request from ABC cards.

6.5.2 Power

It is obvious that a scheme manager has substantial power in an ecosystem. There may often be conflicting interests between verifiers and users. Verifiers want to know as much as possible about users, while users want to release as little information as possible to get access to the verifiers' services. Even though the scheme manager does not participate in transactions, it decides by setting the rules.

User control is essential in a user-centric system, and so it is in an ABC ecosystem. Thus users can initiate issuing and verification procedures, check what credentials and attributes verifiers request. Moreover, ABC cards support users by checking the validity of issuers' and verifiers' certificates to ensure that they do not abuse their power. Nevertheless, users act according to provided possibilities and thus rely heavily on the scheme manager who can decide to limit or overuse attributes on a system level. Users will most often choose to initiate processes when they need to access a particular resource without a thorough check of the verifier. Also, users tend not to be attentive in giving permissions to verifiers, such as authorities and commercial organisations, when they request personal information [AABL13]. Moreover, it is often the case that users do not have a choice in picking a particular verifier or restricting their power. Therefore, a scheme manager's role is paramount in deciding about issuers and verifiers, and their rights. A more elaborate discussion about the socio-technical and legal evaluation of an ABC ecosystem is out of scope in this discussion; see our work for further details [KKAH14].

6.5.3 Principles

This section collects principles that limit the power of a scheme manager by making its working as transparent as possible. We focus on independency, distribution of power and openness.

We firmly believe that the scheme manager should be set up and run as an independent, non-profit and potentially distributed organisation. Since the whole process in relation to attributes and credentials takes place using open standards (and to a large extent even via open-source software), every organisation or individual can use the same card for their own purpose. Therefore, there will be many participants interested in issuing and using their own credentials if they can be based on other (reliable) attributes (*cf.* page 122): probably Facebook, Google, Microsoft (Skype) and possibly also bookshops or supermarket chains. The scheme manager has the technical means (by issuing certificates to issuers and verifiers) to control these processes. Should the scheme manager allow this, and on which grounds? These decisions are political in nature, and they involve the identity fabric of our society and also considerable commercial interest.

To be effective and transparent, an ecosystem has to be open about its operation. We recommend for a scheme manager to publish an 'ecosystem charter' when establishing the ecosystem to set up the basic rules. This document should describe the main objectives of the ecosystem and its principles about selecting members (issuers, verifiers) and defining their capabilities. It should act similarly to a constitution, functioning as the fundament of the laws in a nation state. Such a charter makes it easy to evaluate for issuers and verifiers whether to join. Information about the operation, such as the identity of issuers and verifiers and their permissions in terms of credentials and attributes, should also be public. In this way users or user groups can assess this information and take actions should they disagreed.

We put forth the general principles for an ABC ecosystem and all procedures within it that ensure transparency for the system:

1. The scheme manager is an independent, nonprofit and possibly distributed 'multi-stakeholder'.

2. The design of the whole system and the ecosystem charter are public. Open standards and open-source codes are freely available.

3. The system's security relies solely on the underlying technology and the secret values (private keys and individual attributes).

4. Public-key certificates of issuers and verifiers are publicly accessible for verification.

5. The design of each credential is public (in particular, attribute names in a credential, the issuer's identity and the duration of validity).

6. The issuing procedure in relation to each credential is public (*i.e.* how the authorisation is done before credential issuance); thus, implicitly, credential dependencies are public.

The principles can be classified as the openness of metadata about the ecosystem (1.–3.), the participants (4.) and the credentials (5.–6.).

6.5.4 An Alternative Ecosystem

It is hard to predict the future of attribute-based credentials. We collected possible use cases, proposed an independent scheme manager and laid down principles.

We tried to keep in mind convenience and intuitiveness to stimulate business and governmental interest, while preserving the beneficial security and privacy features of the ABC technology.

We foresee two approaches for the uptake of this technology. This chapter described a top-down approach. Examples for such an ecosystem include a governmental identity system, a public transport card, a health insurance card or a unified commercial loyalty system. In this approach the parties first set up a scheme manager which in turn launches the ecosystem. In a bottom-up approach, on the other hand, different organisations can decide to issue ABC cards that conform to a certain industry standard. It allows arbitrary issuers and service providers to use the platform. The public keys used in such a setup can be arranged in a PKI, or even in a web of trust, and so all participants can verify certificates. In essence in such a setup, a unique scheme manager is not present, rather small groups of stakeholders may decide to create scheme managers of their own and use the open platform to create more closed ABC subsystems (*cf.* Kapor's quote; page 105). Multiple ABC ecosystems then may coexist on a single card. With its open standards and alternative technical solutions, the World Wide Web is perhaps the most important example of a technology that has become widespread with a bottom-up approach.

6.6 Functional Extensions

When an identity system is deployed, revocation and usability are of great importance. Revocation is crucial to build an identity infrastructure in which lost and stolen cards should become unusable. And users should conveniently make use of their ABC cards independently of their location. Without solving these problems, the deployment of such an ecosystem would not be possible or soon fail to become widely used. We briefly discuss these two functions.

6.6.1 Revocation

Revocation of pubic-key certificates is usually done by black- or whitelists which collect the identifiers of all revoked or non-revoked certificates, respectively. However, revocation in the context of Attribute-Based Credentials (and often in privacy-preserving applications in general [LEHS12]) is challenging because there are no identifiers, and in fact, user transactions are unlinkable; see *e.g.* [CL01, CL02, BKMN10, LKDDN11, HM13]. Moreover, in relation to ABC cards, where all the cryptographic computation on the client's side is carried out by a resource-limited smart card, the challenge is even harder. Not only is it required to find the trade-off between anonymity and revocation, but also the client side of the solution has to be efficient enough to run on an ABC card.

In this direction we have done research.[91] Our proposal sets up an ABC ecosystem model where the following assumption holds: The frequency of interactions between users and particular verifiers can be estimated and given a lower bound. For instance, users may want to authorise transactions at the national tax office on a *monthly* basis, but do online shopping at web stores *every day*. To capture this notion, we introduce verifier-specific epochs.

Confirmation of card's validity (*i.e.* the card was not revoked) becomes a part of the verification protocol. A one-way function[92] is specified for the system and each card stores a secret revocation value. The revocation authority keeps a blacklist, that is, a list of the revoked revocation values. For each verifier the revocation authority selects a random generator and creates an epoch-specific list of numbers. Each number is a masked revocation value, computed by the one-way function with the generator and a revocation value as input. The ABC verification process gets slightly extended to enable revocation. When a card is used for authorisation, the verifier's terminal sends not only the above-mentioned certificate with the `Att_Acc`, but also another certificate signed by the revocation authority. Besides other technical components, it contains the generator of the current epoch and the validity period of the generator. The card then provides the effective attribute proofs extended with a value computed by the one-way function with the given generator and the secret revocation value as input. The card also proves, without revealing any secret, that all the computation was performed honestly. The card, not having an internal clock, keeps a time estimate based on the given validity periods of verified certificates. Having received the masked value and the proof from the card, the verifier can look up this value in its epoch-specific list and decide whether the card has been revoked or not.

This method provides security for the verifier and privacy for the user. If a card is revoked (*i.e.* it is on the blacklist of the revocation authority), it cannot avoid being noticed and possibly withdrawn. But a card that is not revoked remains unlinkable as long as it does not revisit the same verifier within the same epoch.

6.6.2 A Mobile Phone as a Personal Card Reader

A smart card is often used with public card readers, such as a cash machine (ATM), a point-of-sale (POS) terminal or an access point. This is also the case with an ABC card. However, there are applications in which the user needs a personal card reader. For instance, if she wants to use her card from home or wishes to manage the card's content. A card reader that is attached to a PC is an obvious choice, but it is not always available, secure or convenient. Therefore, we propose the use of a personal mobile phone that can act as a card reader communicating with an ABC

[91] Joint work with Jaap-Henk Hoepman, Wouter Lueks and Pim Vullers.

[92] A suitable one-way function can be the modular exponentiation in a DL group with a fixed generator; see Section 3.2.1.

card through their contactless interfaces[93]. Since a mobile phone can access the (remote) terminal of the verifier or issuer through the Internet, it can assist remote ABC authorisation processes; see our papers [ABV12, AE13].

The same technology can also help with the card management; see Section 6.3.3. By means of a mobile application, a user can review the content of the card and manage all personal data on it; see an example in Figure 6.3 that shows a tablet screenshot of an Android implementation. Of course, this software has to be protected by the user's administration PIN to prevent illegitimate reading and modification of the card.

By applying a personal mobile phone, users can access all identity services related to their ABC cards. They can authenticate to service providers to access services and they can manage their identity attributes. In this way they become 'location independent' as the 8th law of identity requires, see Section 2.5.1. First, mobile internet access is already virtually ubiquitous and therefore users can access all online services where they can use their ABC cards. They can also authenticate using their phone and the personal ABC card while working on another computer, such as a public PC in an internet café. Second, the card management interface provides identity management functions offline, including deleting obsolete credentials and viewing all attributes on the card. So, users become independent of any external identity services.

6.7 Other Attribute-Based Technologies

There is abundant work on identity management, privacy-enhancing technologies and attribute-based credentials. In Chapter 2 we discussed the main problems in open identity management. In this subsection we focus on attribute-based authorisation techniques, that is, other works in relation to attribute-based credentials and attribute-based access control. We also discuss briefly the relations between these other technologies and our work.

Attributes first appear as a central logical building block in the Attribute-Based Credential technology [Bra00, **?**]. Probably this also affected the then active role-based access control paradigm [SCFY96] that could not satisfy the dynamic and complex access control requirements on the Internet (*e.g.* web services). Attributes as a generalisation of roles (or groups) have been proposed by [WWJ04, YT05].

[93] A detailed description of the technology and the protocols are out of scope here; see details in [ABV12, AE13]. We remark that NFC-enabled mobile phones can communicate with the contactless interface of the ABC card. The requirement that users have personal NFC-enabled phones is realistic since these devices are already prevalent and becoming increasingly popular; see *e.g.* `http://www.nfcworld.com/2014/02/12/327790/two-three-phones-come-nfc-2018/` [last accessed: October 26, 2014].

6.7.1 Attribute-Based Credentials

Since anonymous and attribute-based credentials [Bra00, CL01, **?**, CL04] were devised, much work has been done on possible extensions [CL02, LKDDN11, ABL12, HM13], implementations [SGPV09, BCGS09, TJ09, MV11, VA13], improvements in functionalities [CL02, CG08, CCGS10, ABL12, PAL13, AH13], applications [VLV$^+$08, ABV12, BGOZ12, ALP$^+$12] and possible ABC infrastructures [CVH02, CSFH$^+$05, Bra10, Paq10, Cor11, IBM12, CKL$^+$11, CDL$^+$13, AJ13]. Academic research considers Attribute-Based Credentials one of the most applicable privacy-enhancing technologies and thus invests a lot of efforts to further improve and implement them as well as design new applications and infrastructures based on them. However, in spite of Microsoft's U-Prove[94] and IBM's Idemix[95] open approach, ABCs are not yet employed in practice. A recent project, the ABC4Trust [CKL$^+$11] aims to bridge the gap between the ready-made advanced technology and the lack of interest, by building a common, technology-agnostic framework (*i.e.* swappable U-Prove or Idemix cryptographic components) and demonstrating its usability by pilot projects. In ABC4Trust's view ABCs "will ultimately replace traditional PKI" [SS13], that is, entities will be able to verify universally each other in accordance with the data minimisation principle. Another approach to deploy ABCs can be found within the FutureID project (see Section 6.7.3) that attempts to host several identity technologies within one infrastructure. An ongoing joint effort between the FutureID and the IRMA[96] projects is to incorporate also our approach in this framework.

To embed ABCs in a wider context, credential-based access control [CMN$^+$10] is proposed. It is a general notion that includes role-based and attribute-based access control as well as access control based on complex proofs using attribute-based credentials. Camenisch et al. [CMN$^+$10] propose a language, CARL, for describing requirements that verifiers can present and users have to satisfy with their certificates or credentials. The language is agnostic about the underlying identity management technology. The requirements can be satisfied with various technologies (such as, X.509, SAML, ABCs), and the transaction may or may not include an online identity provider. The abstract language CARL has been developed within the European PrimeLife project, and later it has also been implemented in the XACML access control language [OAS05] (using XML) within the ABC4Trust project [CDL$^+$13].

CARL is a possible choice for defining rules on the verifier's side in an ABC ecosystem. However, we propose first a more straightforward approach when Attribute-Based Credentials are introduced for a wide audience. This would help business and governmental parties to develop simple use cases, and help users to understand the benefits of using an ABC card in terms of simplicity, diverse ap-

[94] `http://research.microsoft.com/en-us/projects/u-prove/`

[95] `http://idemix.wordpress.com`

[96] On IRMA's behalf by Antonio de la Piedra and Jaap-Henk Hoepman

plicability and, most importantly, privacy. Therefore, we propose to apply only a subset of CARL without an XACML/XML implementation that is easy to interpret also for smart cards. A potential approach is that the requirements for selective disclosure proofs are expressed as references to the types of credentials and attributes that can be unique within an ecosystem.

6.7.2 Attribute-Based Access Control

Attribute-Based Access Control (ABAC) [WWJ04, YT05, VFK+14] defines a new and flexible way to define and enforce access policies based on attributes. Attributes are similarly defined as in this thesis but embrace a much bigger set. They can belong not only to a human being but also to a non-person entity. Furthermore, not only the client side but also the resource and even the environment (context) have attributes. The main motivation behind ABAC is to simplify access control in an enterprise context that is first based on identities (like in mandatory and discretionary access control). Later role-based access control [SCFY96] enabled the definition of access policies independently of particular identities. So, first access control abstracts away from identities to roles (groups), and later from roles to attributes. Attributes generalise roles since attributes can describe roles but also many more characteristics.

Claims-based identity management [BBB+11] is a successor of previous works of Microsoft and Kim Cameron [Cam05, CJ07, BSB07]. It aims at developing efficient enterprise IDM with flexible policy evaluation and federated identity management services (such as, single sign-on). Claims-based identity management is closely related to ABAC, but the claims-based approach focusses much more on identity (not only authorisation) and especially on federations of multiple realms. As privacy is not a main concern, claims-based IDM does not satisfy the privacy requirements that attribute-based identity management does. In particular, the requirement of an offline IdP is violated as some attribute service (such as an Active Directory service) is involved in each verification process. Moreover, this component can also link these activities of the user, breaking the unlinkability requirements, because an identifying authentication is required in the same session. ABIdM can be described as a special case of claims-based identity management in which strict privacy restrictions apply.

6.7.3 Related European Projects

Besides ABC4Trust [CKL+11], there are important European projects that we discuss in relation to attribute-based identity management.

The new *German eID* (neuer Personalausweis), introduced in 2010, is a nationwide electronic identity system in which each citizen has an identity card [PWVT12]. Several countries have introduced electronic identity cards, but the German eID is the most advanced working identity card technology in terms of privacy. A card's chip provides three functions: offline identification (ePass), authentication (eID) and electronic signature (eSign). The eID function aims at secure and privacy-friendly authentication online, which is relevant in the context of this thesis. Although the underlying technology is different, there are similarities with an ABC card. First of all, a card can be used for online authentication in a user-centric way.[97] All personal data is stored on the card and can be shown without having an identity provider online. Second, the card verifies the validity of the verifier using its certificate. Also, an end-to-end secure channel established between the service provider's server and the card's chip and all subsequent communication flows through it. Finally, the verifier's certificate describes for the card what this service provider is eligible to request and after the user's consent (including a 6-digit PIN to be entered by the user) the card sends the required information.

There are also important differences that we briefly discuss. In the German model, chip authentication is not only used for setting up a secure channel, but also to establish trust that all data released by such a card inherently originates from the government administration. In terms of ABCs this means that the only attribute issuer is the government. This makes applications much less dynamic, because other issuers cannot participate, unlike in an ABC ecosystem. Another consequence of this technical approach is that attributes in the German model are sent unsigned, and thus the security relies heavily on the smart card. Second, a German eID card is enforced to create a (scope-specific) pseudonym for each transaction. This is not the case in an ABC ecosystem where authentication does not necessarily include a pseudonym, it is just an option (*cf.* Idemix' pseudonyms [IBM12]). Third, there is no card management application that would enable a citizen to manage his personal information stored on his card. The main reason for this is that certificate verification is enforced on the card before every information release instance and only service providers, not clients, can get such certificates. [PWVT12] Finally, revocation in the German model is based on a number distributed to a batch of cards for privacy reasons; see more detail in Section 4.6.2 on page 84. This excludes the possibility to identify specific cards. If a card has to be revoked, this number is placed on a revocation list, and the whole batch needs to be withdrawn. Clearly, this is not practical. Revocation techniques described in [LKDDN11] and the new approach described in Section 6.6.1 show that any card can be revoked in a privacy-friendly way in an ABC ecosystem.

FIDIS (Future of Identity in the Information Society)[98], having run from 2004 to 2009, was a network of excellence project entailing a lot of interdisciplinary research effort in relation to identity. The result of the project is over 120 deliverables that include surveys of current technologies, collections of best practices, proposals to research and management approaches, notes on workshops organised by the

[97] Applications where citizens can use their eID card for authentication purposes are available here: `http://www.personalausweisportal.de/EN/Citizens/Applications/applications_node.html` [last accessed: October 26, 2014]. Apparently, most of these services are identifying.

[98] `http://www.fidis.net`, [last accessed: October 26, 2014]

project, proposals for new techniques, collections of use cases, studies on current practices and policies. Since it covers so many aspects of identity, technology and privacy, FIDIS is relevant in relation to an ABC ecosystem. However, FIDIS does not aim to provide a concrete technology that would provide privacy-friendly authentication comparable to an ABC card or an ABC ecosystem. It does reflect the state of identity management and related research. ABIdM is strongly related to the following FIDIS topics: Privacy and the legal-social content of identity; HighTech ID; Mobility and Identity. What FIDIS recommends for follow-up work is to design and implement a reference architecture and to apply it in important contexts, such as eGovernment, eHealth and mobility [RR09]. Our proposal for an ABC ecosystem is a big step in this direction although it is not concerned about interoperability, federations and complex policies. However, for a well-defined application environment – even if it is so broad as a national government administration – users can manage their identity attributes in a "privacy-aware" manner. The clear notions in ABIdM and its transparency principles (credential design, scheme manager) help identity and service providers as well as users accept this new technology and let this PET get deployed – much in the spirit of FIDIS' results.

STORK (Secure idenTity acrOss boRders linKed)[99] is a technical project that aims to provide a cross-border, interoperable identity service in which citizens can use services using their national electronic identity all over Europe. Potentially, STORK would have a direct relation with our project if a European member state chose to deploy an ABC ecosystem as its national electronic identity. In this case further efforts have to be made towards interoperability with the STORK framework. This includes technical and possibly legal work as well as analysis and risk assessment considering all the relevant issuing processes that in the national ecosystem take place. Conceivably, the interconnection can happen through the FutureID project (see below).

Furthermore, STORK defined four Quality Authentication Assurance (QAA) levels for identification and authentication systems. A thorough analysis is important to carry out in practice with respect to a particular ABC ecosystem. Nevertheless, in principle, authentication in an ABC ecosystem can achieve the highest QAA level (*i.e.* Level 4). This is because of the card and identity provision processes, the protection of authentication against the most important attacks (guessing, eavesdropping, hijacking, replay and man-in-the-middle) and the robustness and security of ABC card authentication.

The *FutureID*[100] project, running from November, 2012 to October, 2015, is a technical project that aims to build an infrastructure that enables authentication throughout Europe independently of the local identity management systems. Thus, it has an umbrella role over most of the eID systems, including STORK. For all three types of participants, *i.e.* IdPs, SP and users, FutureID makes it easy to connect with each other. More specifically, identity providers can connect through an interface

[99] `https://www.eid-stork.eu` and `https://www.eid-stork2.eu` [last accessed: October 26, 2014]

[100] `http://www.futureid.eu` [last accessed: October 26, 2014]

by any standard federation technologies and service providers do not even need to change their current configuration. FutureID intends to incorporate clients of legacy as well as cutting-edge technologies, including Attribute-Based Credentials. In fact, FutureID and the IRMA project are currently cooperating to create an interface that enables ABC cards to communicate with the FutureID platform. As a result, an ABC ecosystem will have an interface and credentials can be issued and verified using the cards.

6.8 Discussion

In this chapter we described attribute-based identity management and realised it by using an ABC card, *i.e.* a full-fledged smart-card implementation of the client side of Attribute-Based Credentials. In this setup users and service providers become independent of actively participating identity providers and any additional user devices.

An ABC ecosystem is a new type of federated system, in which all participants rely on the regulating power of the scheme manager. Each user has an ABC card on which she can collect Attribute-Based Credentials from several identity providers (issuers), and she can proportionately show attributes from them to service providers (verifiers) in various contexts. A scheme manager determines the set of attributes to be revealed specified for each type of resource access. Cards are willing to perform ABC protocols only with legitimate terminals, which is enforced by verifier certificates.

An ABC ecosystem makes it easy for users to understand what personal information they disclose about themselves in a particular situation. But there is a more immediate benefit of our approach (a card implementation, simple attributes, predefined authorisation, *etc.*). As a research project investigating this technology, IRMA showed that it also simplifies comprehension of ABCs for decision-makers in the identity realm. They understand that with the current and upcoming European regulation, ABCs provide a clear way for electronic authentication to comply with the legal framework in terms of processing personal data.

Chapter 7

Conclusion

> "But it was all right, everything was all right, the struggle was finished. He had won the victory over himself."
>
> George Orwell, 1984

This thesis indicates a new and practical approach to the use of attribute-based credentials. It develops the concept of attribute-based identity management (ABIdM) – having its root in security and privacy requirements as well as in an efficient smart-card implementation of ABCs. Authentic attributes, originating from a trusted party about a data subject, enable relying parties to make access decisions to their resources.

In this work we propose to modify the role of identity providers without the need to change the way how identity information is stored at IdPs: They are not involved directly in the authorisation processes. And still, the attributes that users submit to relying parties are up-to-date and authentically related to the users. The identity management, called attribute-based identity management, based on these processes, results in

- reliable data for the service providers (verified, up-to-date and signed data by the identity provider);
- less burden on the identity providers (no required presence at each authentication process); and
- more privacy for the user (no traceability and no dependence on third parties).

Additionally, a user can view all her identity information. The user can easily check attributes and credentials, stored on her personal ABC card, independently of any third parties. Concretely, a card management application is proposed for this purpose.

An ABC ecosystem can provide real privacy to users in various contexts. First of all, it makes it impossible for IdPs and RPs to link a user's activities when she needs only non-identifying attributes for accessing services. By this, users can remain anonymous when identification is not required in a scenario. Second, it also prevents IdPs to sign in on behalf of the user unlike in traditional identity management systems. By this, users can remain in control with regard to their personal data.

Some further technical problems are yet to be solved with regard to the deployment of attribute-based identity management, it can certainly be solved in the near future. First, the security level of the described smart-card implementation is about 80 bits (corresponding to a 1024-bit RSA key). However, this is not considered to be secure enough currently, so it would be desirable to increase it. A later generation of MULTOS cards, having bigger RAM, will probably be able to support a modified implementation of a higher security level. And second, revocation is still hard in the context of attribute-based credentials. Privacy, efficiency and security are hard to align, especially on a restricted infrastructure (*e.g.* no internet connection) and limited client's devices (slow, little computational capabilities, small memory, *etc.*). The revocation technique, described in Section 6.6.1, is a promising way to solve this problem.

In spite of the above-mentioned technical challenges, ABIdM is becoming practical for various applications by the use of secure personal devices, such as smart cards. In principle, high levels of security can be achieved coupled with such privacy guarantees that no other authentication technology can achieve. Therefore, it would be desirable to put this technology in practice. We recognise that this process comprises several important steps: changing business models; raising awareness; creating standards and best practices; and carrying out projects that bring academic studies closer to real-life applications.

To conclude, in Section 7.1 we collect possible research directions and in Section 7.2 we discuss problems that ABIdM solves with respect to the recommendations put forward in Chapter 2.

7.1 Future Research

For further research in the context of digital identity we collected many open issues in the recommendation paragraphs in Chapter 2 and in Section 2.6. Below we offer additional open questions.

Resources, as we discussed them in Section 2.2.2, can be classified by rivalry and durability. Further study would be required with regards to ABC cards being used for accessing different kinds of resources.

As we discussed in Chapter 2, we distinguish federated identity management from non-federated IDM based on the fact whether identities are shared across multiple domains or not. Federated IDM typically involves multiple identity providers and a Directory Service, which acts as a hub among them. In the web of social networks the setup becomes more complex; while traditionally a user has a primary IdP, this is not the case anymore. RPs offer users the possibility to select from several social networks an IdP to access the RP's service. Users have often more than one choice which of their identities (*e.g.* , Google's, Facebook's) to use because they have multiple accounts. In the near future this choice may become harder as customers will have an increasing number of identities, via multiple devices, and they access more and more services. Further research is required to discover what security, privacy and usability consequences this new challenge means for people, what the most optimal choice is and how they can find that particular one in a usable way.

Location independence is a new and essential requirement proposed in Chapter 2. Users should be able to maintain and use their identities independently of the platform they are operating with. Currently, social identity providers (such as, Google, Apple or Facebook) give the possibility for a user to act relying only on some Internet access and a personal account. Note however that this service satisfies only half of the location independence requirement as it violates users' privacy: the identity provider can trace the user and all her activities. ABIdM provides appropriate procedures that allow the users to manage their identities independently of a central IdP and they to authenticate at RPs without involving IdPs in the actual communication. Mainly non-technical research and innovation are required to bridge the gap between this technology and practical deployment in the social context. Finding viable solutions probably requires to re-design business models, to introduce new policies and to raise awareness amongst users and business owners about potential benefits of ABIdM.

Further research is required to analyse to what extent can a designated verifier proof be a substitute for a secure channel. This research includes the analysis of many aspects. (1) Efficiency: Establishing a secure channel may involve the use of public-key cryptography, but confidentiality and authenticity of messages rely on efficient symmetric-key primitives. Designated verifier proofs, however, extended zero-knowledge proofs that perform purely especially on resource-constrained devices. (2) Security model: Since the two techniques are very different in nature, it is required to take particular attention when comparing their security. (3) Functionality: A secure channel provides flexibility with regard to messages that it conveys, while designated verifier proofs are much more rigid in structure.

A possible research project can define the achievable QAA levels (see page 133) using ABC cards. This certainly depends not only the cryptographic and implementation parameters, but also organisational, infrastructural and contextual decisions within an ABC ecosystem.

7.2 Discussion

Finally, we discuss the main benefits of an ABC ecosystem that ameliorate the current identity crisis. An ABC ecosystem (Chapter 6) is an (attribute-based) identity management system, and as such, we can discuss whether it implements recommendations we put forward in Chapter 2.

- Dynamic identity. An ABC card provides the possibility for users to collect their attributes from several issuers. Credentials and sets of credentials can be considered as partial identities of the user that can be issued and shown under the user's control. In diverse contexts users can show different yet authentic attributes about themselves to verifiers. Nevertheless, because of technical limitations on the speed of issuance and verification, an ABC card is not yet suitable for applications in which very short-lived identities and tokens are required, such as public digital transport tickets, electronic wallet functions, *etc.*

- Phishing attacks. There are two important technical measures in an ABC ecosystem that prevent phishing attacks. First, there is no digitally conveyed secret, like a password, that can be stolen. A user (or her card) sends only zero-knowledge proofs as authentication of attributes, which cannot be reused by an adversary. Second, IdPs and RPs have to authenticate to the card before they can communicate with it. Thus, mutual authentication is mandated. The technique does not require additional user interaction.

- User privacy. ABIdM inherits all privacy properties defined at ABCs; see Section 3.3 on page 52. Therefore, users need to reveal only the minimum amount of personal information for a given service and this happens without interacting with IdPs (credential issuers). Moreover, although users have to rely on the scheme manager that authentication instances happen in the most privacy-friendly way, they can still practice control by reviewing the requested attributes and giving their consent before actual data disclosure.

- Location Independence. Having an ABC card, users can make use of all features that the ABC ecosystem provides, and they can interact with identity and service providers without having to rely on their location or devices. Thus, the system satisfies the 8th Law of Identity, about Location Independence (see page 34).

- Determine minimal role automatically. In an ABC ecosystem the scheme manager is responsible to decide about the minimal amount of revealed attributes that users need for accessing a given resource. Relying on these decisions, this requirement is satisfied from the user's point of view when the actual authentication occurs.

- Federation and user control. An ABC ecosystem can be considered as a special type of user-centric federation, in which several issuers provide attributes that can be used for authorisation purposes in multiple organisational domains (at verifiers). However, SSO is not provided since sessions are intrinsically unrelated.

Security and privacy are historically trade-offs in the digital world. The more it is known what users are doing, the more secure the system is often considered to be – and in turn, the less privacy the users have. But this compromise is not a fundamental law that systems are bound to comply with. Rather it is a technical legacy that we are conditioned to live with. PETs are demonstrating that security and privacy can exist simultaneously, and in fact, they can even reinforce each other. ABCs, their smart-card implementation and an ABC ecosystem, for instance, decrease the necessity of storing a lot of personal data at various parties, and by this, it increases the security of the overall system. Additionally, it gives users control over the disclosure of their personal information and ultimately, it enhances trust in the technology and achieves a higher level of acceptance.

Bibliography

[AABL13] I. Adjerid, A. Acquisti, L. Brandimarte, and G. Loewenstein. Sleights of privacy: framing, disclosures, and the limits of transparency. In *Proceedings of the Ninth Symposium on Usable Privacy and Security (SOUPS '12)*, page 9. ACM, 2013. (1 citation: Page 125.)

[ABL12] G. Alpár, L. Batina, and W. Lueks. Designated Attribute-Based Proofs for RFID Applications. In J.-H. Hoepman and I. Verbauwhede, editors, *RFID Security and Privacy – RFIDsec 2012*, LNCS 7739, pages 59–75. Springer, 2012. (2 citations: Pages 11 and 130.)

[ABV12] G. Alpár, L. Batina, and R. Verdult. Using NFC Phones for Proving Credentials. In J. B. Schmitt, editor, *Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, LNCS 7201, pages 317–330. Springer, 2012. (2 citations: Pages 129 and 130.)

[AE13] G. Alpár and M. H. Everts. Mobile devices to the identity rescue. In *IFIP Summer School on Privacy and Identity Management for Emerging Services and Technologies, Nijmegen, The Netherlands, June 17-21, 2013, Revised Selected Papers*, IFIP AICT 421, pages 243–253. Springer, 2013. (1 citation: Page 129.)

[AH13] G. Alpár and J.-H. Hoepman. A secure channel for attribute-based credentials:[short paper]. In *Proceedings of the 2013 ACM workshop on Digital identity management (DIM)*, pages 13–18. ACM, 2013. (2 citations: Pages 10 and 130.)

[AHS13] G. Alpár, J.-H. Hoepman, and J. Siljee. The Identity Crisis – Security, Privacy and Usability Issues in Identity Management. *Journal of Information System Security*, 9(1):23–53, 2013. (1 citation: Page 9.)

[AJ13] G. Alpár and B. Jacobs. Credential design in attribute-based identity management. In *Bridging distances in technology and regulation, 3rd TILTing Perspectives Conference*, pages 189–204, 2013. (2 citations: Pages 11 and 130.)

[ALP$^+$12] J. Abendroth, V. Liagkou, A. Pyrgelis, C. Raptopoulos, A. Sabouri, E. Schlehahn, Y. Stamatiou, and H. Zwingelberg. D7.1 Application Description for Students. Technical report, ABC4Trust, 2012. (2 citations: Pages 107 and 130.)

[AM07] W. A. Alrodhan and C. J. Mitchell. Addressing privacy issues in CardSpace. In *3rd International Symposium on Information Assurance and Security (IAS 2007)*, pages 285–291. IEEE, 2007. (1 citation: Page 20.)

[And08] R. Anderson. *Security engineering*. Wiley. com, 2008. (1 citation: Page 34.)

[BA12] L. Brandimarte and A. Acquisti. *The Economics of Privacy*. The Oxford Handbook of the Digital Economy. Oxford University Press, 2012. (1 citation: Page 20.)

[Bac05] J. Backhouse. D4. 1: Structured account of approaches on interoperability. *FIDIS Deliverables*, 2005. (1 citation: Page 108.)

[Bal08] J. Balasch. Smart Card Implementation of Anonymous Credentials. Master's thesis, Katholieke Universiteit Leuven, 2008. (2 citations: Pages 59 and 63.)

[BBB$^+$11] D. Baier, V. Bertocci, K. Brown, S. Densmore, E. Pace, and M. Woloski. *A Guide to Claims-Based Identity and Access Control: Authentication and Authorization for Services and the Web*. Microsoft patterns & practices (online edition: `http://msdn.microsoft.com/en-us/library/ff423674.aspx`), 2nd edition, September 2011. (2 citations: Pages 16 and 131.)

[BC11] F. Bélanger and R. E. Crossler. Privacy in the Digital Age: A Review of Information Privacy Research in Information Systems. *MIS Quarterly*, 35(4):1017–1041, 2011. (1 citation: Page 14.)

[BCC04] E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145. ACM, 2004. (2 citations: Pages 63 and 84.)

[BCGS09] P. Bichsel, J. Camenisch, T. Gross, and V. Shoup. Anonymous credentials on a standard Java Card. In *Computer and Communications Security (CCS 2009)*, pages 600–610. ACM, November 2009. (3 citations: Pages 59, 63, and 130.)

[BCI08] J. Bringer, H. Chabanne, and T. Icart. Cryptanalysis of EC-RAC, a RFID identification protocol. In M. K. Franklin, L. C. K. Hui, and D. S. Wong, editors, *Cryptology and Network Security – CANS 2008*, LNCS, pages 149–161. Springer, 2008. (5 citations: Pages 89, 91, 97, 101, and 102.)

[BFK09] J. Bender, M. Fischlin, and D. Kügler. Security analysis of the PACE key-agreement protocol. In *Information Security*, pages 33–48. Springer, 2009. (1 citation: Page 84.)

[BGOZ12] S. Bcheri, N. Goetze, M. Orski, and H. Zwingelberg. D6.1 Application Description for the School Deployment. Technical report, ABC4Trust, 2012. (2 citations: Pages 107 and 130.)

[Bic07] P. Bichsel. Theft and misuse protection for anonymous credentials. Master's thesis, ETH Zürich, Switzerland, 2007. (1 citation: Page 59.)

[BKMN10] J. Bender, D. Kügler, M. Margraf, and I. Naumann. Privacy-friendly revocation management without unique chip identifiers for the German national ID card. *Computer Fraud & Security*, September 2010. (1 citation: Page 127.)

[BKPR14] R. Bjones, I. Krontiris, P. Paillier, and K. Rannenberg. Integrating anonymous credentials with eIDs for privacy-respecting online authentication. In *Privacy Technologies and Policy*, pages 111–124. Springer, 2014. (2 citations: Pages 59 and 105.)

[BL12] F. Baldimtsi and A. Lysyanskaya. Anonymous credentials light. *IACR Cryptology ePrint Archive*, 2012:298, 2012. (2 citations: Pages 47 and 53.)

[BLS$^+$11] L. Batina, Y. K. Lee, S. Seys, D. Singelée, and I. Verbauwhede. Privacy-preserving ECC-based grouping proofs for RFID. In *Information Security*, pages 159–165. Springer, 2011. (1 citation: Page 89.)

[BM10] N. Bohm and S. Mason. Identity and its verification. *Computer Law & Security Review*, 26(1):43–51, 2010. (1 citation: Page 21.)

[BMH05] M. Bauer, M. Meints, and M. Hansen. D3.1: Structured Overview on Prototypes and Concepts of Identity Management Systems. Technical report, Future of Identity in the Information Society (FIDIS), 2005. (3 citations: Pages 15, 21, and 107.)

[BP97] N. Barić and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology—EUROCRYPT'97*, pages 480–494. Springer, 1997. (1 citation: Page 44.)

[BR94] M. Bellare and P. Rogaway. Entity authentication and key distribution. In *Advances in Cryptology—CRYPTO'93*, pages 232–249. Springer, 1994. (2 citations: Pages 63 and 66.)

[Bra94] S. Brands. Untraceable off-line cash in wallet with observers. In *Advances in Cryptology—CRYPTO'93*, pages 302–318. Springer, 1994. (1 citation: Page 51.)

[Bra00] S. A. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, Cambridge, MA, USA, 2000. (12 citations: Pages 3, 9, 17, 20, 42, 46, 47, 48, 53, 106, 129, and 130.)

[Bra10] S. Brands. U-Prove technology overview. Technical report, Microsoft Corporation, March 2010. (6 citations: Pages 17, 33, 42, 53, 58, and 130.)

[BSB07] V. Bertocci, G. Serack, and C. Baker. *Understanding Windows CardSpace: An introduction to the concepts and challenges of digital identities*. Pearson Education, 2007. (1 citation: Page 131.)

[BSCGS07] A. Bhargav-Spantzel, J. Camenisch, T. Gross, and D. Sommer. User centricity: a taxonomy and open issues. *Journal of Computer Security*, 15(5):493–527, 2007. (4 citations: Pages 28, 105, 114, and 117.)

[BSSV12] L. Batina, S. Seys, D. Singelée, and I. Verbauwhede. Hierarchical ECC-based RFID authentication protocol. In *RFID. Security and Privacy*, pages 183–201. Springer, 2012. (2 citations: Pages 89 and 103.)

[Cam05] K. Cameron. The laws of identity. *Microsoft Corporation*, 2005. (5 citations: Pages 13, 18, 28, 34, and 131.)

[Cav06] A. Cavoukian. *7 laws of identity: The case for privacy-embedded laws of identity in the digital age*. Information and Privacy Commissioner/Ontario, 2006. (2 citations: Pages 13 and 15.)

[CCGS10] J. Camenisch, N. Casati, T. Gross, and V. Shoup. Credential authenticated identification and key exchange. In *Advances in Cryptology–CRYPTO 2010*, pages 255–276. Springer, 2010. (2 citations: Pages 85 and 130.)

[CDL$^+$13] J. Camenisch, M. Dubovitskaya, A. Lehmann, G. Neven, C. Paquin, and F.-S. Preiss. Concepts and languages for privacy-preserving attribute-based authentication. In *Policies and Research in Identity Management (IDMAN)*, pages 34–52. Springer, 2013. (1 citation: Page 130.)

[CG08] J. Camenisch and T. Gross. Efficient attributes for anonymous credentials. In *Proceedings of the 15th ACM conference on Computer and communications security (CCS 2008)*, pages 345–356. ACM, 2008. (1 citation: Page 130.)

[Cha83] D. Chaum. Blind signatures for untraceable payments. In D. Chaum and R. L. Rivest, editors, *Advances in Cryptology – CRYPTO 1982*, pages 199–203. Plemum Publishing, 1983. (1 citation: Page 51.)

[Cha85] D. Chaum. Security without identification: transaction systems to make big brother obsolete. *Communications of the ACM*, 28:1030–1044, October 1985. (1 citation: Page 83.)

[Cha91] D. Chaum. Zero-knowledge undeniable signatures. In *Advances in Cryptology—EUROCRYPT'90*, pages 458–464. Springer, 1991. (1 citation: Page 88.)

[CJ07] K. Cameron and M. B. Jones. Design rationale behind the identity metasystem architecture. In *ISSE/SECURE 2007 Securing Electronic Business Processes*, pages 117–129. Springer, 2007. (2 citations: Pages 19 and 131.)

[CKL$^+$11] J. Camenisch, I. Krontiris, A. Lehmann, G. Neven, C. Paquin, K. Rannenberg, and H. Zwingelberg. D2.1 Architecture for Attribute-based Credential Technologies. Technical report, ABC4Trust, 2011. (6 citations: Pages 17, 33, 83, 107, 130, and 131.)

[CL01] J. Camenisch and A. Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In B. Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer Berlin / Heidelberg, 2001. (10 citations: Pages 3, 9, 17, 20, 42, 61, 83, 106, 127, and 130.)

[CL02] J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In M. Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *LNCS*, pages 101–120. Springer, August 2002. (2 citations: Pages 127 and 130.)

[CL04] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology–CRYPTO 2004*, pages 56–72. Springer, 2004. (2 citations: Pages 53 and 130.)

[CM07] D. Cvrček and V. Matyáš. D13.1: Identity and impact of privacy enhancing technology. Technical report, Future of Identity in the Information Society (FIDIS), 2007. (1 citation: Page 20.)

[CMN$^+$10] J. Camenisch, S. Mödersheim, G. Neven, F.-S. Preiss, and D. Sommer. A card requirements language enabling privacy-preserving access control. In *Proceedings of the 15th ACM symposium on Access control models and technologies*, pages 119–128. ACM, 2010. (1 citation: Page 130.)

[Cor11] F. Corella. On the prospects for using privacy-enhancing technologies in the NSTIC identity ecosystem, http://pomcor.com/2011/10/04/pros-and-cons-of-u-prove-for-nstic/. Blog, October 2011. (1 citation: Page 130.)

[CP09] P. Carpenter and E. Perkins. Magic quadrant for user provisioning. Gartner, September 2009. (1 citation: Page 15.)

[CPR09] K. Cameron, R. Posch, and K. Rannenberg. Proposal for a common identity framework: A user-centric identity metasystem. *The Future of Identity in the Information Society (FIDIS) – Appendix D*, page 477, 2009. (2 citations: Pages 21 and 114.)

[CS97] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In *Advances in Cryptology—CRYPTO'97*, pages 410–424. Springer, 1997. (2 citations: Pages 50 and 65.)

[CSFH$^+$05] J. Camenisch, D. Sommer, S. Fischer-Hübner, M. Hansen, H. Krasemann, G. Lacoste, R. Leenes, J. Tseng, et al. Privacy and identity management for everyone. In *Proceedings of workshop on Digital Identity Management (DIM), Fairfax, VA, USA*, pages 20–27. ACM, 2005. (3 citations: Pages 21, 105, and 130.)

[CVA90] D. Chaum and H. Van Antwerpen. Undeniable signatures. In *Advances in Cryptology—CRYPTO'89 Proceedings*, pages 212–216. Springer, 1990. (1 citation: Page 88.)

[CVH91] D. Chaum and E. Van Heyst. Group signatures. In *Advances in Cryptology—EUROCRYPT'91*, pages 257–265. Springer, 1991. (1 citation: Page 84.)

[CVH02] J. Camenisch and E. Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Computer and Communications Security (CCS 2002)*, pages 21–30. ACM, November 2002. (2 citations: Pages 107 and 130.)

[Dam99] I. Damgård. Commitment schemes and zero-knowledge protocols. In *Lectures on Data Security*, pages 63–86. Springer, 1999. (1 citation: Page 46.)

[Daw06] R. Dawkins. *The selfish gene.* Oxford University Press, 2006. (1 citation: Page 39.)

[DD08] R. Dhamija and L. Dusseault. The seven flaws of identity management: Usability and security challenges. *Security & Privacy, IEEE*, 6(2):24–29, 2008. (4 citations: Pages 19, 20, 28, and 29.)

[DF02] I. Damgård and E. Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *Advances in Cryptology—ASIACRYPT 2002*, pages 125–142. Springer, 2002. (1 citation: Page 49.)

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on*, 22(6):644–654, 1976. (1 citation: Page 81.)

[DH08] P. De Hert. Identity management of e-ID, privacy and security in Europe – A human rights view. *Information Security Technical Report*, 13(2):71–75, 2008. (1 citation: Page 19.)

[dlPHV14] A. de la Piedra, J.-H. Hoepman, and P. Vullers. Towards a full-featured implementation of Attribute Based Credentials on smart cards. In D. Gritzalis, A. Kiayias, and I. Askoxylakis, editors, *13th Int. Conf. on Cryptology and Network Security (CANS 2014)*, LNCS 8813, pages 270–289. Springer International Publishing, 2014. (1 citation: Page 110.)

[DM97] S. Deakin and J. Michie. *Contracts, co-operation, and competition: studies in economics, management, and law.* Oxford University Press, 1997. (1 citation: Page 25.)

[Ell99] C. M. Ellison. The nature of a useable PKI. *Computer Networks*, 31(8):823–830, 1999. (2 citations: Pages 30 and 34.)

[Eur95] European Parliament and the European Council. 'Data Protection' Directive 95/46/EC. `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML`, November 1995. (4 citations: Pages 19, 23, 33, and 105.)

[FH02] S. Farrell and R. Housley. RFC 3281: An Internet Attribute Certificate Profile for Authorization, April 2002. (1 citation: Page 106.)

[FH07] D. Florencio and C. Herley. A large-scale study of web password habits. In *Proceedings of the 16th international conference on World Wide Web*, pages 657–666. ACM, 2007. (1 citation: Page 5.)

[Fin14] K. Finklea. Identity Theft: Trends and Issues. Technical Report 7-5700, R40599, Congressional Research Service, 2014. (2 citations: Pages 13 and 29.)

[FO97] E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology—CRYPTO'97*, pages 16–30. Springer, 1997. (2 citations: Pages 44 and 49.)

[FS87] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. Odlyzko, editor, *Advances in Cryptology – CRYPTO '86*, volume 263 of *LNCS*, pages 186–194. Springer, 1987. (4 citations: Pages 51, 65, 75, and 88.)

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989. (2 citations: Pages 50 and 88.)

[Har04] R. Hardin. *Trust and trustworthiness*, volume 4. Russell Sage Foundation, 2004. (1 citation: Page 25.)

[HM13] J. Hajny and L. Malina. Unlinkable attribute-based credentials with practical revocation on smart-cards. In *Smart Card Research and Advanced Applications*, pages 62–76. Springer, 2013. (3 citations: Pages 53, 127, and 130.)

[HWF09] D. Hein, J. Wolkerstorfer, and N. Felber. ECC is Ready for RFID – A Proof in Silicon. In *Selected Areas in Cryptography – SAC 2008*, pages 401–413. Springer, 2009. (1 citation: Page 89.)

[IBM12] IBM Research, Security Team. Specification of the Identity Mixer Cryptographic Library, version 2.3.4. Technical report, IBM Research, Zürich, February 2012. (11 citations: Pages 17, 33, 42, 44, 49, 53, 55, 83, 107, 130, and 132.)

[ISO11] ISO/IEC. 14443-3:2011 Identification Cards – Contactless Integrated Circuit(s) Cards – Proximity Cards. Part 3: Initialization and Anticollision, 2011. (1 citation: Page 115.)

[JCB09] D.-O. Jaquet-Chiffelle and H. Buitelaar. D17. 4: Trust and Identification in the Light of Virtual Persons. Technical report, Future of Identity in the Information Society (FIDIS), 2009. (1 citation: Page 25.)

[JSI96] M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In *Advances in Cryptology—EUROCRYPT'96*, pages 143–154. Springer, 1996. (1 citation: Page 88.)

[Jue04] A. Juels. “Yoking-proofs” for RFID tags. In *Pervasive Computing and Communications Workshops, 2004. Proceedings*, pages 138–143. IEEE, 2004. (1 citation: Page 89.)

[JWD08] R. Joosten, D. Whitehouse, and P. Duquenoy. Towards a meta model for identity terminology. In *Pre-proceedings of the IFIP/FIDIS Internet Security & Privacy Summer School*, pages 141–146, Masaryk University, Brno, Czech Republic, September 1–7, 2008, 2008. (1 citation: Page 14.)

[JZS07] A. Jøsang, M. A. Zomai, and S. Suriadi. Usability and Privacy in Identity Management Architectures. In *ACSW Frontiers*, pages 143–152, 2007. (2 citations: Pages 14 and 20.)

[KKAH14] M. Koning, P. Korenhof, G. Alpár, and J.-H. Hoepman. The ABCs of ABCs: an analysis of attribute-based credentials in the light of data protection, privacy and identity. In *Balcells, J., Cerrillo i Martínez, A., Peguera, M., Peña-López, I., Pifarré de Moner, MJ, & Vilasau Solana, M.(coords.)(2014). A decade of transformations. Proceedings of the 10th International Conference on Internet, Law & Politics. Universitat Oberta de Catalunya, Barcelona*, pages 357–374, 2014. (1 citation: Page 125.)

[KL08] J. Katz and Y. Lindell. *Introduction to modern cryptography*. CRC Press, 2008. (1 citation: Page 44.)

[KO02] M. Koch and D. Of. Global identity management to boost personalization. In *Proc. Research Symposium on Emerging Electronic Markets*, pages 137–147, 2002. (1 citation: Page 14.)

[Kob87] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987. (1 citation: Page 89.)

[Kos14] E. Kosta. Personal communication, May 2014. (1 citation: Page 19.)

[Lam74] B. W. Lampson. Protection. *ACM SIGOPS Operating Systems Review*, 8(1):18–24, 1974. (1 citation: Page 31.)

[LBSV10] Y. K. Lee, L. Batina, D. Singelée, and I. Verbauwhede. Low-cost untraceable authentication protocols for RFID. In *Proceedings of the third ACM conference on Wireless network security*, pages 55–64. ACM, 2010. (3 citations: Pages 87, 89, and 102.)

[LCP06] H. Lacohée, S. Crane, and A. Phippen. Trustguide: Final report. *Trustguide. October*, 2006. (1 citation: Page 25.)

[LEHS12] W. Lueks, M. H. Everts, J.-H. Hoepman, and R. J. Siljee. Revocable Privacy 2011 – use cases. Technical Report 35627, TNO, 2012. (1 citation: Page 127.)

[Lin07] Y. Lindell. Anonymous authentication. *Journal of Privacy and Confidentiality*, 2(2):4, 2007. (1 citation: Page 84.)

[LKDDN11] J. Lapon, M. Kohlweiss, B. De Decker, and V. Naessens. Analysis of revocation strategies for anonymous Idemix credentials. In *Communications and Multimedia Security*, pages 3–17. Springer, 2011. (3 citations: Pages 127, 130, and 132.)

[LLVGW09] S. Landau, H. Le Van Gong, and R. Wilton. Achieving privacy in a federated identity management system. In *Financial Cryptography and Data Security*, pages 51–70. Springer, 2009. (2 citations: Pages 19 and 20.)

[LRSW00] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. In *Selected Areas in Cryptography*, pages 184–199. Springer, 2000. (1 citation: Page 83.)

[Mal06] J. Malinen. Windows CardSpace. Technical report, Helsinki University of Technology, 2006. (2 citations: Pages 17 and 26.)

[MDPDD14] M. Milutinovic, I. Dacosta, A. Put, and B. De Decker. An advanced, privacy-friendly loyalty system. In *IFIP Summer School on Privacy and Identity Management for Emerging Services and Technologies, Nijmegen, The Netherlands, June 17-21, 2013, Revised Selected Papers*, IFIP AICT 421, pages 133–143. Springer, 2014. (1 citation: Page 111.)

[MKGV07] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007. (2 citations: Pages 22 and 109.)

[MM12] J. R. Mayer and J. C. Mitchell. Third-party web tracking: Policy and technology. In *Security and Privacy (SP), 2012 IEEE Symposium on*, pages 413–427. IEEE, 2012. (1 citation: Page 111.)

[MOVR97] A. J. Menezes, P. C. V. Oorschot, S. A. Vanstone, and R. L. Rivest. Handbook of applied cryptography, 1997. (1 citation: Page 44.)

[MR08] E. Maler and D. Reed. The Venn of Identity: Options and Issues in Federated Identity Management. *IEEE Security & Privacy*, 6(2):16–23, 2008. (2 citations: Pages 17 and 20.)

[MV11] W. Mostowski and P. Vullers. Efficient U-Prove implementation for anonymous credentials on smart cards. In G. Kesidis and H. Wang, editors, *Security and Privacy in Communication Networks – SecureComm 2011*, volume 96 of *LNICST*, pages 243–260. Springer, 2011. (5 citations: Pages 10, 43, 59, 63, and 130.)

[MVS08] G. Moniava, E. Verheul, and L. Schoenmakers. Extending DigiD to the private sector (DigiD-2). Master's thesis, Department of Mathematics and Computing Science, Eindhoven University of Technology, 2008. (1 citation: Page 32.)

[Nis04] H. Nissenbaum. Privacy as Contextual Integrity. *Washington Law Review*, 79(1):119–158., February 2004. (1 citation: Page 23.)

[OAS05] OASIS. eXtensible Access Control Markup Language (XACML) Version 2.0. `http://docs.oasis-open.org/xacml/`, February 2005. (1 citation: Page 130.)

[OEC80] OECD. Guidelines governing the protection of privacy and transborder flows of personal data. Technical report, OECD, 1980. (1 citation: Page 120.)

[O'H04] K. O'Hara. *Trust: from Socrates to spin*. Icon Books, 2004. (1 citation: Page 25.)

[OHF07] B. Otjacques, P. Hitzelberger, and F. Feltz. Interoperability of e-government information systems: Issues of identification and data sharing. *Journal of Management Information Systems*, 23(4):29–51, 2007. (1 citation: Page 122.)

[OT12] D. O'Brien and A. M. Torres. Social Networking and Online Privacy: Facebook Users' Perceptions. *Irish Journal of Management*, 31(2):63–97, 2012. (1 citation: Page 20.)

[PAL13] K. Papagiannopoulos, G. Alpár, and W. Lueks. Designated attribute proofs with the Camenish–Lysyanskaya signature. In *Proceedings of the 34rd WIC Symposium on Information Theory in the Benelux, Leuven, Belgium. May 30—31, 2013.*, 2013. (2 citations: Pages 104 and 130.)

[Paq10] C. Paquin. U-Prove technology integration into the identity metasystem v1.0. Technical report, Microsoft Corporation, March 2010. (1 citation: Page 130.)

[Paq11] C. Paquin. U-Prove Cryptographic Specification v1.1. Technical report, Microsoft Corporation, February 2011. (2 citations: Pages 53 and 83.)

[PBP10] A. Pfitzmann and K. Borcea-Pfitzmann. Lifelong privacy: Privacy and identity management for life. In *Privacy and Identity Management for Life*, pages 1–17. Springer, 2010. (1 citation: Page 20.)

[Pea09] S. Pearson. Taking account of privacy when designing cloud computing services. In *Software Engineering Challenges of Cloud Computing, 2009. CLOUD'09. ICSE Workshop on*, pages 44–52. IEEE, 2009. (1 citation: Page 20.)

[Ped92] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology—CRYPTO'91*, pages 129–140. Springer, 1992. (1 citation: Page 47.)

[PH10] A. Pfitzmann and M. Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. *Version 0.34 Aug*, 10, 2010. (2 citations: Pages 19 and 107.)

[PM03] A. Pashalidis and C. J. Mitchell. A taxonomy of single sign-on systems. In *Information security and privacy*, pages 249–264. Springer, 2003. (1 citation: Page 15.)

[PWVT12] A. Poller, U. Waldmann, S. Vowé, and S. Türpe. Electronic identity cards for user authentication—promise and practice. *IEEE Security & Privacy*, 10(1):46–54, 2012. (4 citations: Pages 32, 84, 131, and 132.)

[RBB03] P. Rogaway, M. Bellare, and J. Black. OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Transactions on Information and System Security (TISSEC)*, 6(3):365–403, 2003. (1 citation: Page 78.)

[Roy13] D. Royer. *Enterprise Identity Management: Towards an Investment Decision Support Approach.* Springer, 2013. (1 citation: Page 14.)

[RR09] K. Rannenberg and D. Royer. Open Challenges – Towards the (Not So Distant) Future of Identity. In *The Future of Identity in the Information Society*, pages 391–399. Springer Berlin Heidelberg, 2009. (1 citation: Page 133.)

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. (1 citation: Page 44.)

[RST01] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *Advances in Cryptology—ASIACRYPT 2001*, pages 552–565. Springer, 2001. (1 citation: Page 84.)

[SCFY96] R. S. Sandhu, E. J. Coynek, H. L. Feinsteink, and C. E. Youmank. Role-based access control models. *IEEE computer*, 29(2):38–47, 1996. (2 citations: Pages 129 and 131.)

[Sch91] C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of cryptology*, 4(3):161–174, 1991. (3 citations: Pages 50, 51, and 88.)

[Sch11] H. A. Schmidt. The National Strategy for Trusted Identities in Cyberspace. `http://www.whitehouse.gov/sites/default/files/rss_viewer/NSTICstrategy_041511.pdf`, April 2011. (2 citations: Pages 13 and 21.)

[SGPV09] M. Sterckx, B. Gierlichs, B. Preneel, and I. Verbauwhede. Efficient implementation of anonymous credentials on Java Card smart cards. In *Information Forensics and Security – WIFS 2009*, pages 106–110. IEEE, September 2009. (2 citations: Pages 59 and 130.)

[SJ10] J. Scudder and A. Jøsang. Personal federation control with the identity dashboard. In *Policies and Research in Identity Management*, pages 85–99. Springer, 2010. (1 citation: Page 15.)

[SKM04] S. Saeednia, S. Kremer, and O. Markowitch. An efficient strong designated verifier signature scheme. In *Information Security and Cryptology-ICISC 2003*, pages 40–54. Springer, 2004. (1 citation: Page 88.)

[SS13] P. Spirakis and Y. C. Stamatiou. Attribute based credentials towards refined public consultation results and effective egovernance. In *Cyber Security and Privacy*, pages 115–126. Springer, 2013. (1 citation: Page 130.)

[Swe02] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002. (2 citations: Pages 22 and 109.)

[TECA07] J. Tsai, S. Egelman, L. Cranor, and A. Acquisti. The effect of online privacy information on purchasing behavior: An experimental study. In *ICIS 2007 Proceedings*, 2007. (1 citation: Page 20.)

[TFM05] J. Turow, L. Feldman, and K. Meltzer. Open to exploitation: American shoppers online and offline. Technical report, Annenberg Public Policy Center, University of Pennsylvania, Philadelphia, PA, 2005. (1 citation: Page 20.)

[TJ09] H. Tews and B. Jacobs. Performance issues of selective disclosure and blinded issuing protocols on Java Card. In O. Markowitch, A. Bilas, J.-H. Hoepman, C. Mitchell, and J.-J. Quisquater, editors, *Information Security Theory and Practice – WISTP 2009*, volume 5746 of *LNCS*, pages 95–111. Springer, September 2009. (2 citations: Pages 59 and 130.)

[VA13] P. Vullers and G. Alpár. Efficient selective disclosure on smart cards using Idemix. In S. Fischer-Hübner, E. de Leeuw, and C. Mitchell, editors, *Policies and Research in Identity Management (IDMAN)*, pages 53–67. Springer, 2013. (7 citations: Pages 10, 33, 43, 59, 63, 105, and 130.)

[Vau06] S. Vaudenay. RFID privacy based on public-key cryptography. In *Information Security and Cryptology – ICISC 2006*, pages 1–6. Springer, 2006. (1 citation: Page 89.)

[Vau07] S. Vaudenay. On privacy models for RFID. In *Advances in Cryptology – ASIACRYPT 2007*, pages 68–87. Springer, 2007. (3 citations: Pages 87, 89, and 97.)

[VDWZ11] M. Veeningen, B. De Weger, and N. Zannone. Modeling identity-related properties and their privacy strength. In *Formal Aspects of Security and Trust (FAST*

2011), Leuven (Belgium), September 12-14, pages 126–140. Springer, 2011. (1 citation: Page 19.)

[Ver01] E. R. Verheul. Self-blindable credential certificates from the weil pairing. In *Advances in cryptology—ASIACRYPT 2001*, pages 533–551. Springer, 2001. (4 citations: Pages 3, 50, 53, and 106.)

[VFK$^+$14] C. H. Vincent, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone. Guide to Attribute Based Access Control (ABAC) Definition and Considerations. NIST Special Publication 800-162, National Institute of Standards and Technology, January 2014. (2 citations: Pages 106 and 131.)

[VLV$^+$08] K. Verslype, J. Lapon, P. Verhaeghe, V. Naessens, and B. De Decker. PetAnon: A privacy-preserving e-petition system based on Idemix. CW Reports CW522, Department of Computer Science, K.U.Leuven, October 2008. (1 citation: Page 130.)

[Vul14] P. Vullers. *Efficient Implementations of Attribute-based Credentials on Smart Cards*. PhD thesis, Radboud University Nijmegen, 2014. (2 citations: Pages 59 and 105.)

[WWJ04] L. Wang, D. Wijesekera, and S. Jajodia. A logic-based framework for attribute based access control. In *Proceedings of the 2004 ACM workshop on Formal methods in security engineering*, pages 45–55. ACM, 2004. (3 citations: Pages 106, 129, and 131.)

[YT05] E. Yuan and J. Tong. Attributed based access control (ABAC) for web services. In *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*. IEEE, 2005. (3 citations: Pages 106, 129, and 131.)

Index

8th law of identity, *see* location independence

ABC, *see* attribute-based credential
ABC scheme, 64
ABC4Trust, 21, 107, 130
ABCDH, *see* Diffie–Hellman ABC Channel
ABIdM, *see* identity management
access, 24
access control, 112, 129, 130
Amazon, 5
anonymity, 61
anonymous authentication, 84
Apple, 5, 137
attribute, 40, 107
attribute point, 93
attribute-based
  access control, 107, 131
  credential, 40, 107, 129–131
  credential scheme, 64
  identity management, 114, 134
  public-key credential, 81
authorisation, 116

benign adversary, 68
binding, 75
blind signature, 51
BSN, 32

CA, *see* certificate authority
CAID, 85
CAKE, 85
card
  management, 65, 118, 129, 135
  provisioning, 115
card management, 118
card provider, 114
CARL, 130–131
certificate, 65
certificate authority, 33, 65, 124, 125
claim, 16
claims-based identity management, 131
combine, 69
commitment, 46
confidential key setup, 69
corrupt adversary, 68
CPA-security, 73
credential
  access credential, 112
  age credential, 109
  citizen credential, 110
  credential design, 119–123
  credential tree, 120
  design principles, 123
  festival ticket, 113
  login credential, 113
  loyalty card, 110
  medical credential, 111
  phone number, 112
CRM, *see* customer relationship management
customer relationship management, 111

DAA, *see* direct anonymous attestation
DDH problem, *see* decisional Diffie–Hellman problem
decisional Diffie–Hellman problem, 90
dependent credential, 120
designated verifier, 91
  identification, 92
  proof, 88, 137
  selective disclosure, 93–102

designated verifier selective disclosure, 94
Diffie–Hellman ABC Channel, 79–82
DigiD, 32
direct anonymous attestation, 63, 84
disclosing set, 53, 90
discrete logarithm
  assumption, 44
  group, 44
  problem, 44, 89
DL, *see* discrete logarithm
domain, 15
DPD, 19, 105
DSD, *see* designated verifier selective disclosure
durability, 25, 136

EC, *see* elliptic-curve
ecosystem, 105, 123, 126, 136, 138
  charter, 124, 126
elliptic-curve
  discrete logarithm problem, 89
  representation, 90
  representation problem, 90
empty proof, 57, 59, 65, 121
encryption, 71
entitlement set, 93
entity, 14
equality proof, 62

Facebook, 5, 20, 23, 37, 108, 122–123, 137
Fiat–Shamir heuristic, 51
FIDIS, 21, 132–133
FutureID, 130, 133–134

GDPR, 19, 105
generalised Pedersen commitment, 47
German eID, 84, 131–132
Google, 5, 31, 37, 137

Higgins, 21

ICA, *see* implicit card authentication
Idemix, 54, 57, 59, 130
identifier, 14
identity, 14, 18, 138
  crisis, 13, 37, 138
  laws of, 18
  partial, 107, 138
  provider, 15, 135, 137
  provisioning, 15
identity management, 13, 15
  attribute-based, 114, 134–139
  claims-based, 131
  enterprise, xxiii
  federated, 14, 15, 17, 20, 26, 27, 30, 31, 131, 137, 138
  user-centric, 114
implementation, 59–60
implicit card authentication, 70–78
indistinguishability, 67
IRMA, 6, 63, 107
issuer, 40, 91, 114
  offline, 52
issuing, 53, 54, 116
  unlinkability, 4, 58

key confirmation, 71

LinkedIn, 37
location independence, 33, 34, 137, 138
login, 113

matching conversation, 67, 68
membership, 24
Microsoft, 17, 19, 23, 30
minimal information, 53, 138
mobile phone, 112, 128
multi-show unlinkability, 4
MULTOS, 59, 63, 136
mutual authentication, 20, 38, 61, 85

narrow-strong privacy, 98
NoMatch, 69
non-repudiation, 52
non-transferability, 52

ownership, 24

PACE, 84
partial identity, 107
Pedersen commitment, 43, 46–48, 89
PET, 31, 133
phishing, 38, 138
PIN, 114, 117, 118, 132
  administration, 118
PKI, 33, 127, 130

privacy, 20, 31–34, 100, 135, 138, 139
protocol, 70
pseudonym, 110, 132

QAA levels, 133, 137

RAM, 63, 136
Randomized Schnorr, 89, 91, 92, 101
representation, 45, 90
    assumption, 46
representation problem, 45, 90
revocation, 69, 84, 85, 127–128, 132
RFID, 87–88
    tag, 91
rivalry, 25, 136
role, 35, 138
root credential, 65, 120
RSA problem, 44
    strong, 45

scheme manager, 123–127
Schnorr proof, 50, 51, 54, 89
SD, *see* selective disclosure
secret key, 40, 114
secure ABC authentication, 67, 70
secure binding, 76
secure channel, 61–85
security, 20, 27–31, 72, 98, 139
selective disclosure, 4, 41, 51, 52, 56–58, 65
signature, 48
    randomisation, 50
single sign-on, 20, 30, 35, 138
smart card, 59, 61
social login, 5, 137
SSO, *see* single sign-on
STORK, 133
strong RSA problem, 45
surveillance, 28

trust, 25

U-Prove, 53, 56, 59, 130
uncorrupted, 68
unforgeability, 52
unlinkability, 58, 97
    issuer, 52
    multi-show, 52
usability, 20, 34–37
user, 114
user control, 18, 114, 117, 125, 138

valid, 63, 66
verification, *see also* selective disclosure, 116
verifier, 40, 114, 135

zero-knowledge proof, 50, 51, 88, 95

Abbreviations

ABAC	Attribute-Based Access Control
ABC	Attribute-Based Credential
ABIdM	attribute-based identity management
ABPKC	attribute-based public-key credential
BSN	Burgerservicenummer
CA	Certificate Authority
CDH	Computational Diffie–Hellman
CL	Camenisch–Lysyanskaya
CoT	Circle of Trust
CRM	Customer Relationship Management
DAA	Direct Anonymous Attestation
DDH	Decisional Diffie–Hellman
DH	Diffie–Hellman
DID	designated verifier identification
DL	discrete logarithm
DS	Directory Service
DSD	designated verifier selective disclosure
ECC	elliptic-curve cryptography
EC	elliptic curve
ESTA	Electronic System for Travel Authorization
EU	European Union
FIDIS	Future of Identity in the Information Society
ICT	Information and Communications Technology
IDM	identity management
IdP	identity provider

IMS	identity management system
IRMA	I Reveal My Attributes
ISP	Internet service provider
MAC	Message Authentication Code
MNO	mobile network operator
NIZK	non-interactive zero-knowledge
OS	operating system
PACE	Password Authenticated Connection Establishment
PET	privacy-enhancing technology
PIN	Personal Identification Number
PKC	public-key cryptography
PKI	public-key infrastructure
RP	relying party
RFID	Radio-Frequency IDentification
SD	selective disclosure
SOA	service-oriented architecture
SP	service provider
SSN	social security number
SSO	single sign-on
U	user
UA	user agent
WAYF	Where-Are-You-From

Curriculum vitae

Gergely Alpár

1974 Born in Budapest, Hungary

1994 - 2000 Master of Science
Mathematics and Mathematics Education
Eötvös Loránd University, Budapest, Hungary

2000 - 2002 Master of Technological Design[101]
Mathematics for Industry, Communication and Information Processes orientation
Eindhoven University of Technology, The Netherlands

2002 - 2010 Director
Amega Team Ltd.,[102] Budapest, Hungary

2010 - 2014 Ph.D.
Identity Management for Mobile Devices
Radboud University, Nijmegen, The Netherlands

2014 - Researcher and Teacher
Radboud University, Nijmegen, The Netherlands

[101] The degree later got renamed: Professional Doctorate in Engineering

[102] Online communication and IT services company

Titles in the IPA Dissertation Series since 2009

M.H.G. Verhoef. *Modeling and Validating Distributed Embedded Real-Time Control Systems.* Faculty of Science, Mathematics and Computer Science, RU. 2009-01

M. de Mol. *Reasoning about Functional Programs: Sparkle, a proof assistant for Clean.* Faculty of Science, Mathematics and Computer Science, RU. 2009-02

M. Lormans. *Managing Requirements Evolution.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-03

M.P.W.J. van Osch. *Automated Model-based Testing of Hybrid Systems.* Faculty of Mathematics and Computer Science, TU/e. 2009-04

H. Sozer. *Architecting Fault-Tolerant Software Systems.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-05

M.J. van Weerdenburg. *Efficient Rewriting Techniques.* Faculty of Mathematics and Computer Science, TU/e. 2009-06

H.H. Hansen. *Coalgebraic Modelling: Applications in Automata Theory and Modal Logic.* Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2009-07

A. Mesbah. *Analysis and Testing of Ajax-based Single-page Web Applications.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-08

A.L. Rodriguez Yakushev. *Towards Getting Generic Programming Ready for Prime Time.* Faculty of Science, UU. 2009-9

K.R. Olmos Joffré. *Strategies for Context Sensitive Program Transformation.* Faculty of Science, UU. 2009-10

J.A.G.M. van den Berg. *Reasoning about Java programs in PVS using JML.* Faculty of Science, Mathematics and Computer Science, RU. 2009-11

M.G. Khatib. *MEMS-Based Storage Devices. Integration in Energy-Constrained Mobile Systems.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-12

S.G.M. Cornelissen. *Evaluating Dynamic Analysis Techniques for Program Comprehension.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2009-13

D. Bolzoni. *Revisiting Anomaly-based Network Intrusion Detection Systems.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-14

H.L. Jonker. *Security Matters: Privacy in Voting and Fairness in Digital Exchange.* Faculty of Mathematics and Computer Science, TU/e. 2009-15

M.R. Czenko. *TuLiP - Reshaping Trust Management.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-16

T. Chen. *Clocks, Dice and Processes.* Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2009-17

C. Kaliszyk. *Correctness and Availability: Building Computer Algebra on top of Proof Assistants and making Proof Assistants available over the Web.* Faculty of Science, Mathematics and Computer Science, RU. 2009-18

R.S.S. O'Connor. *Incompleteness & Completeness: Formalizing Logic and Analysis in Type Theory.* Faculty of Science, Mathematics and Computer Science, RU. 2009-19

B. Ploeger. *Improved Verification Methods for Concurrent Systems.* Faculty of Mathematics and Computer Science, TU/e. 2009-20

T. Han. *Diagnosis, Synthesis and Analysis of Probabilistic Models*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-21

R. Li. *Mixed-Integer Evolution Strategies for Parameter Optimization and Their Applications to Medical Image Analysis*. Faculty of Mathematics and Natural Sciences, UL. 2009-22

J.H.P. Kwisthout. *The Computational Complexity of Probabilistic Networks*. Faculty of Science, UU. 2009-23

T.K. Cocx. *Algorithmic Tools for Data-Oriented Law Enforcement*. Faculty of Mathematics and Natural Sciences, UL. 2009-24

A.I. Baars. *Embedded Compilers*. Faculty of Science, UU. 2009-25

M.A.C. Dekker. *Flexible Access Control for Dynamic Collaborative Environments*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2009-26

J.F.J. Laros. *Metrics and Visualisation for Crime Analysis and Genomics*. Faculty of Mathematics and Natural Sciences, UL. 2009-27

C.J. Boogerd. *Focusing Automatic Code Inspections*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2010-01

M.R. Neuhäußer. *Model Checking Nondeterministic and Randomly Timed Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2010-02

J. Endrullis. *Termination and Productivity*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2010-03

T. Staijen. *Graph-Based Specification and Verification for Aspect-Oriented Languages*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2010-04

Y. Wang. *Epistemic Modelling and Protocol Dynamics*. Faculty of Science, UvA. 2010-05

J.K. Berendsen. *Abstraction, Prices and Probability in Model Checking Timed Automata*. Faculty of Science, Mathematics and Computer Science, RU. 2010-06

A. Nugroho. *The Effects of UML Modeling on the Quality of Software*. Faculty of Mathematics and Natural Sciences, UL. 2010-07

A. Silva. *Kleene Coalgebra*. Faculty of Science, Mathematics and Computer Science, RU. 2010-08

J.S. de Bruin. *Service-Oriented Discovery of Knowledge - Foundations, Implementations and Applications*. Faculty of Mathematics and Natural Sciences, UL. 2010-09

D. Costa. *Formal Models for Component Connectors*. Faculty of Sciences, Division of Mathematics and Computer Science, VUA. 2010-10

M.M. Jaghoori. *Time at Your Service: Schedulability Analysis of Real-Time and Distributed Services*. Faculty of Mathematics and Natural Sciences, UL. 2010-11

R. Bakhshi. *Gossiping Models: Formal Analysis of Epidemic Protocols*. Faculty of Sciences, Department of Computer Science, VUA. 2011-01

B.J. Arnoldus. *An Illumination of the Template Enigma: Software Code Generation with Templates*. Faculty of Mathematics and Computer Science, TU/e. 2011-02

E. Zambon. *Towards Optimal IT Availability Planning: Methods and Tools*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-03

L. Astefanoaei. *An Executable Theory of Multi-Agent Systems Refinement*. Faculty of Mathematics and Natural Sciences, UL. 2011-04

J. Proença. *Synchronous coordination of distributed components*. Faculty of Mathematics and Natural Sciences, UL. 2011-05

A. Moralı. *IT Architecture-Based Confidentiality Risk Assessment in Networks of Organizations.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-06

M. van der Bijl. *On changing models in Model-Based Testing.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-07

C. Krause. *Reconfigurable Component Connectors.* Faculty of Mathematics and Natural Sciences, UL. 2011-08

M.E. Andrés. *Quantitative Analysis of Information Leakage in Probabilistic and Nondeterministic Systems.* Faculty of Science, Mathematics and Computer Science, RU. 2011-09

M. Atif. *Formal Modeling and Verification of Distributed Failure Detectors.* Faculty of Mathematics and Computer Science, TU/e. 2011-10

P.J.A. van Tilburg. *From Computability to Executability – A process-theoretic view on automata theory.* Faculty of Mathematics and Computer Science, TU/e. 2011-11

Z. Protic. *Configuration management for models: Generic methods for model comparison and model co-evolution.* Faculty of Mathematics and Computer Science, TU/e. 2011-12

S. Georgievska. *Probability and Hiding in Concurrent Processes.* Faculty of Mathematics and Computer Science, TU/e. 2011-13

S. Malakuti. *Event Composition Model: Achieving Naturalness in Runtime Enforcement.* Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2011-14

M. Raffelsieper. *Cell Libraries and Verification.* Faculty of Mathematics and Computer Science, TU/e. 2011-15

C.P. Tsirogiannis. *Analysis of Flow and Visibility on Triangulated Terrains.* Faculty of Mathematics and Computer Science, TU/e. 2011-16

Y.-J. Moon. *Stochastic Models for Quality of Service of Component Connectors.* Faculty of Mathematics and Natural Sciences, UL. 2011-17

R. Middelkoop. *Capturing and Exploiting Abstract Views of States in OO Verification.* Faculty of Mathematics and Computer Science, TU/e. 2011-18

M.F. van Amstel. *Assessing and Improving the Quality of Model Transformations.* Faculty of Mathematics and Computer Science, TU/e. 2011-19

A.N. Tamalet. *Towards Correct Programs in Practice.* Faculty of Science, Mathematics and Computer Science, RU. 2011-20

H.J.S. Basten. *Ambiguity Detection for Programming Language Grammars.* Faculty of Science, UvA. 2011-21

M. Izadi. *Model Checking of Component Connectors.* Faculty of Mathematics and Natural Sciences, UL. 2011-22

L.C.L. Kats. *Building Blocks for Language Workbenches.* Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2011-23

S. Kemper. *Modelling and Analysis of Real-Time Coordination Patterns.* Faculty of Mathematics and Natural Sciences, UL. 2011-24

J. Wang. *Spiking Neural P Systems.* Faculty of Mathematics and Natural Sciences, UL. 2011-25

A. Khosravi. *Optimal Geometric Data Structures.* Faculty of Mathematics and Computer Science, TU/e. 2012-01

A. Middelkoop. *Inference of Program Properties with Attribute Grammars, Revisited.* Faculty of Science, UU. 2012-02

Z. Hemel. *Methods and Techniques for the Design and Implementation of Domain-Specific Languages*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2012-03

T. Dimkov. *Alignment of Organizational Security Policies: Theory and Practice*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2012-04

S. Sedghi. *Towards Provably Secure Efficiently Searchable Encryption*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2012-05

F. Heidarian Dehkordi. *Studies on Verification of Wireless Sensor Networks and Abstraction Learning for System Inference*. Faculty of Science, Mathematics and Computer Science, RU. 2012-06

K. Verbeek. *Algorithms for Cartographic Visualization*. Faculty of Mathematics and Computer Science, TU/e. 2012-07

D.E. Nadales Agut. *A Compositional Interchange Format for Hybrid Systems: Design and Implementation*. Faculty of Mechanical Engineering, TU/e. 2012-08

H. Rahmani. *Analysis of Protein-Protein Interaction Networks by Means of Annotated Graph Mining Algorithms*. Faculty of Mathematics and Natural Sciences, UL. 2012-09

S.D. Vermolen. *Software Language Evolution*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2012-10

L.J.P. Engelen. *From Napkin Sketches to Reliable Software*. Faculty of Mathematics and Computer Science, TU/e. 2012-11

F.P.M. Stappers. *Bridging Formal Models – An Engineering Perspective*. Faculty of Mathematics and Computer Science, TU/e. 2012-12

W. Heijstek. *Software Architecture Design in Global and Model-Centric Software Development*. Faculty of Mathematics and Natural Sciences, UL. 2012-13

C. Kop. *Higher Order Termination*. Faculty of Sciences, Department of Computer Science, VUA. 2012-14

A. Osaiweran. *Formal Development of Control Software in the Medical Systems Domain*. Faculty of Mathematics and Computer Science, TU/e. 2012-15

W. Kuijper. *Compositional Synthesis of Safety Controllers*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2012-16

H. Beohar. *Refinement of Communication and States in Models of Embedded Systems*. Faculty of Mathematics and Computer Science, TU/e. 2013-01

G. Igna. *Performance Analysis of Real-Time Task Systems using Timed Automata*. Faculty of Science, Mathematics and Computer Science, RU. 2013-02

E. Zambon. *Abstract Graph Transformation – Theory and Practice*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2013-03

B. Lijnse. *TOP to the Rescue – Task-Oriented Programming for Incident Response Applications*. Faculty of Science, Mathematics and Computer Science, RU. 2013-04

G.T. de Koning Gans. *Outsmarting Smart Cards*. Faculty of Science, Mathematics and Computer Science, RU. 2013-05

M.S. Greiler. *Test Suite Comprehension for Modular and Dynamic Systems*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2013-06

L.E. Mamane. *Interactive mathematical documents: creation and presentation*. Faculty of Science, Mathematics and Computer Science, RU. 2013-07

M.M.H.P. van den Heuvel. *Composition and synchronization of real-time components upon one processor*. Faculty of Mathematics and Computer Science, TU/e. 2013-08

J. Businge. *Co-evolution of the Eclipse Framework and its Third-party Plug-ins*. Faculty of Mathematics and Computer Science, TU/e. 2013-09

S. van der Burg. *A Reference Architecture for Distributed Software Deployment*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2013-10

J.J.A. Keiren. *Advanced Reduction Techniques for Model Checking*. Faculty of Mathematics and Computer Science, TU/e. 2013-11

D.H.P. Gerrits. *Pushing and Pulling: Computing push plans for disk-shaped robots, and dynamic labelings for moving points*. Faculty of Mathematics and Computer Science, TU/e. 2013-12

M. Timmer. *Efficient Modelling, Generation and Analysis of Markov Automata*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2013-13

M.J.M. Roeloffzen. *Kinetic Data Structures in the Black-Box Model*. Faculty of Mathematics and Computer Science, TU/e. 2013-14

L. Lensink. *Applying Formal Methods in Software Development*. Faculty of Science, Mathematics and Computer Science, RU. 2013-15

C. Tankink. *Documentation and Formal Mathematics — Web Technology meets Proof Assistants*. Faculty of Science, Mathematics and Computer Science, RU. 2013-16

C. de Gouw. *Combining Monitoring with Run-time Assertion Checking*. Faculty of Mathematics and Natural Sciences, UL. 2013-17

J. van den Bos. *Gathering Evidence: Model-Driven Software Engineering in Automated Digital Forensics*. Faculty of Science, UvA. 2014-01

D. Hadziosmanovic. *The Process Matters: Cyber Security in Industrial Control Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-02

A.J.P. Jeckmans. *Cryptographically-Enhanced Privacy for Recommender Systems*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-03

C.-P. Bezemer. *Performance Optimization of Multi-Tenant Software Systems*. Faculty of Electrical Engineering, Mathematics, and Computer Science, TUD. 2014-04

T.M. Ngo. *Qualitative and Quantitative Information Flow Analysis for Multi-threaded Programs*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-05

A.W. Laarman. *Scalable Multi-Core Model Checking*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-06

J. Winter. *Coalgebraic Characterizations of Automata-Theoretic Classes*. Faculty of Science, Mathematics and Computer Science, RU. 2014-07

W. Meulemans. *Similarity Measures and Algorithms for Cartographic Schematization*. Faculty of Mathematics and Computer Science, TU/e. 2014-08

A.F.E. Belinfante. *JTorX: Exploring Model-Based Testing*. Faculty of Electrical Engineering, Mathematics & Computer Science, UT. 2014-09

A.P. van der Meer. *Domain Specific Languages and their Type Systems*. Faculty of Mathematics and Computer Science, TU/e. 2014-10

B.N. Vasilescu. *Social Aspects of Collaboration in Online Software Communities*. Faculty of Mathematics and Computer Science, TU/e. 2014-11

F.D. Aarts. *Tomte: Bridging the Gap between Active Learning and Real-World Systems*. Faculty of Science, Mathematics and Computer Science, RU. 2014-12

N. Noroozi. *Improving Input-Output Conformance Testing Theories*. Faculty of Mathematics and Computer Science, TU/e. 2014-13

M. Helvensteijn. *Abstract Delta Modeling: Software Product Lines and Beyond*. Faculty of Mathematics and Natural Sciences, UL. 2014-14

P. Vullers. *Efficient Implementations of Attribute-based Credentials on Smart Cards*. Faculty of Science, Mathematics and Computer Science, RU. 2014-15

F.W. Takes. *Algorithms for Analyzing and Mining Real-World Graphs*. Faculty of Mathematics and Natural Sciences, UL. 2014-16

M.P. Schraagen. *Aspects of Record Linkage*. Faculty of Mathematics and Natural Sciences, UL. 2014-17

G. Alpár. *Attribute-Based Identity Management: Bridging the Cryptographic Design of ABCs with the Real World*. Faculty of Science, Mathematics and Computer Science, RU. 2015-01