CERIAS Tech Report 2007-84
Protocols and Systems for Privacy Preserving Protection of Digital Identity
by Abhilasha Bhargav-Spantzel
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Graduate School ETD Form 9
(01/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis Acceptance

This is to certify that the thesis prepared

By Abhilasha Bhargav-Spantzel

Entitled

Protocols and Systems for Privacy Preserving Protection of Digital Identity

Complies with University regulations and meets the standards of the Graduate School for originality and quality

For the degree of Doctor of Philosophy

Final examining committee members

Elisa Bertino, Chair

Eugene H. Spafford

Ninghui Li

Sunil Prabhakar

Approved by Major Professor(s): Elisa Bertino

Approved by Head of Graduate Program: Aditya Mathur / William J. Gorman

Date of Graduate Program Head's Approval: 26 October 2007

PROTOCOLS AND SYSTEMS FOR PRIVACY PRESERVING PROTECTION OF DIGITAL IDENTITY

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Abhilasha Bhargav-Spantzel

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2007

Purdue University

West Lafayette, Indiana

To my wonderful family

असतो मा सद्गमय ।

तमसो मा ज्योतिर्गमय ।

मृत्योर्माऽमृतं गमय ॥

ACKNOWLEDGMENTS

I am very fortunate to have a great family and wonderful friends, colleagues, teachers and mentors to whom I am deeply grateful.

I thank my advisor, Elisa Bertino, for her guidance, patience and encouragement over these years. If I became a better writer and researcher in my time as a Ph.D. student, it is largely because of her. She gave me an opportunity to pursue my research interests, and taught me to think outside the box and have the potential of creating a broad impact. She has also taught me to always continue with energy and vigor despite the problems that we face in research.

I would like to express my gratitude to Prof. Eugene H. Spafford, Prof. Ninghui Li and Prof. Sunil Prabhakar for their guidance as my advisory committee. Thanks also to Prof. Christopher Clifton and Prof. Cristina Nita-Rotaru for their help in my Ph.D. qualifying process. Special thanks to Prof. Samuel S. Wagstaff who first got me interested in research as an undergraduate student, and showed me how much fun research is. I also thank Prof. Jeffery M. Siskind and Prof. Robert W. Proctor for wonderful opportunities to pursue undergraduate research.

I am thankful to numerous members of the center for education and research in information assurance and security (CERIAS) for their friendship, help and family-like support. I am also thankful to the Purdue Computer Science Department for the excellent education and opportunities.

I would like to thank my friends and colleagues with whom I had some excellent research and study opportunities: Vamsi Vytla, Pranay Gupta, Wei Jiang, Anna Squicciarini, Rui Xue, Dieter Sommer, Thomas Gross, Jan Camenisch, Xiangwei Kong, Weike Zhang and Jungha Woo.

I would like to thank my friends Ashish Kamra, Siddhartha Arora who helped me revise my introductory dissertation chapters. In addition I extend my thanks to Ashish Kamra,

Anna Squicciarini, Muralikrishna Ramanathan , Jayesh Pandey, Jacques Daniel Thomas, Ashish Kundu, Mohamed Yoosuf Mohamed Nabeel and Wei Jiang for helping me revising my final dissertation presentation.

I am very thankful to my friends and mentors from the industry: Symantec, Liberty Alliance, IBM and Intel. Special thanks to Wes Higaki from Symantec Corporation for his continued guidance and support. Thanks also to Venkateswaran Padmanabhan at Intel for a great learning experience during my internship. I am thankful to Rampalli Prasad and Ketan Sampat from Intel, for fascinating and exciting projects and opportunities as I take my next step into the real world.

My deepest gratitude is to my wonderful family which is my lifetime driving force and essence of happiness, purpose, achievement and contentment. Their unconditional love and closeness to my everyday life, makes it easier to face difficult times and relish each other's successes. Beyond any expression of words, I prize each member of my family where each individual is a fighter when it comes to facing difficult problems with a positive attitude, and at the same time enjoys life to the fullest. I have tried to learn various qualities from each member of my family.

My father, Mr. Ashok Kumar Bhargava who has a strong sense of what is practical, and how to organize things while taking care of critical details. I have always seen him with the best smile even during the hardest times. It's with a good sense of humor and vigorous persistence and perseverance that I see him face any situation. My father has always guided me to focus and excel on my studies ever since I started. He came online on an almost daily basis to talk about current activities, and be there for us with advice, comfort or simply fun, all of which is something we treasure a lot. My mother, Mrs. Vandana Bharrgava who redefines what it is to be ambitious; she opens the mind to believe in aiming high with no restrictions, and following it up with diligent and tireless pursuance of the goals. She is the elegant embodiment of excellence and knowledge along with simplicity. Right from the beginning of my college education, my mother has fought out the wide variety of problems we had; right from obtaining the visa, getting airline tickets, arranging for tuition and so on, to make any of this possible.

My big brother, Mr. Abhinav Bhargava whom I have tried to copy so many times ever since childhood, especially in times when I sought for energy to deal with people who weren't necessarily doing the right thing. He has always been protective of everyone. He excels in anything he takes up because of his honest passion and natural dedication to the job at hand. Unlike most youngsters in the current generation, he does not hesitate to face the real hard problems related to society; it would take a person like him to make significant progress even to the most underdeveloped parts of India. My sister-in-law, Mahima Bhargava, a recent member of our family, and who (along with her wonderful family) has added new happiness in her sweet and simple hearted ways and a cheerful attitude. My little sister, Vidisha Bhargava who simply is the joy of the family, and is enthusiastic to rejoice at any given opportunity. I learn from her not only to work well with people, and take initiative but also to enjoy every moment with a cheerful attitude. Vidisha has been more than a helping hand in various ways throughout my college life, with her selflessly taking care of more things than I have acknowledged or realized. She has always taken care ever since she learnt how to talk, and she learnt that really well ◡̈.

I thank my husband's parents Thomas Spantzel and Eva Maria Spantzel, and sister Anne-Kathrin Spantzel for their kind encouragement and support. The chocolate packages from Germany always helped in lifting spirits and staying energized for work. I would like to thank all my grandparents who's love and guidance will stay with me always. I would also like to thank my uncle Amar Priyadarshee and my cousin Anand Priyadarshee for their help and support throughout my college studies.

I am especially grateful to *meinen liebsten Ehemann*, Joerg Spantzel, who has been by my side each moment, giving the strength, encouragement, love and support, without which I do not think I would be able to achieve any of this. He is patient and provides me with solace even in the most anxious moments. He is my best friend and everything is fine, once he is around. He has also helped me tremendously with my research work; we brainstormed together and he also helped me polish my research papers. Last few months he has helped me revise each chapter of my dissertation and prepare my final presentation. Overall Joerg makes life beautiful every single day.

TABLE OF CONTENTS

Page

Page

Page

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

IDM	Identity Management
IdP	Identity Provider
SP	Service Provider
ZKPK	Zero Knowledge Proof of Knowledge
SVD	Singular Value Decomposition
SVM	Support Vector Machines
PKI	Public Key Infrastructure

ABSTRACT

Bhargav-Spantzel, Abhilasha Ph.D., Purdue University, December, 2007. Protocols and Systems for Privacy Preserving Protection of Digital Identity. Major Professor: Elisa Bertino.

To support emerging online activities within the digital information infrastructure, such as commerce, healthcare, entertainment and scientific collaboration, it is increasingly important to verify and protect the digital identity of the individuals involved. Identity management systems manage the digital identity life cycle of individuals which includes issuance, usage and revocation of digital identifiers.

Identity management systems have improved the management of identity information and user convenience; however they do not provide specific solutions to address protection of identity from threats such as identity theft and privacy violation. One major shortcoming of current approaches is the lack of strong verification techniques for issuance and usage of digital identifiers. In the absence of verification mechanisms, digital identifiers can be misused to commit identity theft. Another shortcoming is the inability of individuals to disclose minimal data while satisfying strong identity verification requirements. The extraneous data collected can potentially be aggregated or used in a manner that would lead to violation of an individual's privacy. Finally, current identity management systems do not consider biometric and history-based identifiers. Such identifiers are increasingly becoming an integral part of an individual's identity. Such types of identity data also need to be used with other digital identifiers and protected against misuse.

In this thesis we introduce a number of techniques that address the above problems. Our approach is based on the concept of privacy preserving multi-factor identity verification. The technique consists of verifying multiple identifier claims of an individual, without revealing extraneous identity information. A distinguishing feature of our approach is that

we employ identity protection and verification techniques in all stages of the identity life cycle. We also enhance our approach with the use of biometric and history-based identifiers. In particular we provide the following key contributions:

- A new cryptographic primitive referred to as *aggregate proof of knowledge* to achieve privacy preserving multi-factor verification. This primitive uses aggregate signatures on commitments that are then used for aggregate zero-knowledge proof of knowledge (ZKPK) protocols. Our cryptographic scheme is better in terms of the performance, flexibility and storage requirements than existing efficient ZKPK techniques that may be used to prove, under zero-knowledge, the knowledge of multiple secrets.

- Algorithms to generate biometric keys reliably from an individual's biometric images. These keys are used to create biometric commitments that are subsequently used to perform multi-factor identity verification using ZKPK. Several factors, including various traditional identity attributes, can thus be used in conjunction with one or more biometrics of the individual. We also ensure security and privacy of the biometric data and show how the biometric key is not revealed even if all the data, including cryptographic secrets, stored at the client machine are compromised.

- A series of protocols for the establishment and management of an individual's transaction history-based identifiers encoded as receipts from e-commerce transactions. These receipt protocols satisfy the security and privacy requirements related to the management of the electronic receipts. We also demonstrate how the users receipt protocols can be employed in the context of mobile phones. In particular we provide techniques to manage the portable identity information on such devices, and use them at physical locations of the service providers.

1. INTRODUCTION

1.1 Motivations

The emerging information infrastructure connects remote parties worldwide through the use of large scale networks, and through a diverse and complex set of software technologies. Activities in various domains, such as commerce, entertainment, scientific collaboration, healthcare and so forth, are increasingly being carried out based on the use of remote resources and services. These resources and services are engaged at various levels within those domains. The interaction between different parties at remote locations may be (and sometimes should be) based on only little knowledge about each other.

To better support these activities and collaborations, Information Technology (IT) infrastructure and systems are needed that are more convenient to use. We expect, for example, that personal preferences and profiles of individuals be readily available when shopping over the Internet or when running jobs on a computing grid, without requiring the individuals to repeatedly enter them. In such a scenario, digital identity management (IdM) technology is fundamental in customizing user experience, underpinning accountability in transactions, and complying with regulatory controls. For this technology to fully deploy its potential, it is crucial that strong *protection of digital identity* be achieved. IdM systems must assure that such information is not misused and individuals' privacy is guaranteed. The goal of the work reported in this thesis is to devise solutions to the problem of identity theft and misuse in IdM systems. We develop a fundamental approach to the problem by focusing on multi-factor identity verification, nonetheless preserving the privacy of the individuals. Our approach to digital identifiers goes beyond the traditional identifiers, for example social security numbers, to also include biometric identifiers and individuals' history of online activities. In what follows, we introduce relevant background information and elaborate on the motivations and the goals of our research.

1.1.1 Digital Identity

Digital identity can be defined as the digital representation of the information known about a specific individual or organization. More specifically, our notion of digital identity refers to two different, not necessarily disjoint, concepts: nyms and partial identities. A *nym* gives an individual an identity under which to operate when interacting with other parties; an example of a nym is a login name or a pseudonym. Nyms can be strongly bound or linked to an individual, or be meaningful only in the context of a specific application domain. Weakly bound or unbound nyms are useful in contexts such as chat rooms and on-line games. *Partial identities* encompass a set of properties, such as name, birth date, credit-cards-numbers, patient-record-number, which are referred to as *identity attributes* or *identifiers*, that are associated with individuals. We use an identity attribute as a synonym of identifier. Each subset of identifiers represents the partial identity of the individual. Partial identities may or may not be bound to the human identity of one or more actual individuals.

It is important to note the issue of *identity ownership* as the identity attributes of individuals are stored and shared among various entities in IdM systems. By *owner of an identity attribute* we mean the individual to whom this identity attribute is issued to by a trusted authority or an individual who is authoritative with respect to the claiming of the identifier. In the former case, the trusted issuer of the identifier is also responsible for providing information about the *validity* of that identifier. Validity of an identifier encompasses several notions (some of which are derived from the field of data quality [1]); examples of such notions are: correctness, that is, the identifier is correct (possibly with respect to the real-world); timeliness, that is, the identifier is up to date.

When talking about identifiers, it is also important to distinguish between weak and strong identifiers. A strong identifier uniquely identifies an individual in a population, whereas a weak identifier can be applied to many individuals in a population. Whether an identifier is strong or weak depends upon the size of the population and the uniqueness of the identity attribute. The combination of multiple weak identifiers may lead to a unique identification [2, 3]. Examples of strong identifiers are an individual's passport number or

social security number (SSN). Weak identifiers are attributes such as citizenship, age and gender. This distinction is significant because misuse of strong identifiers can have more serious consequences, such as identity theft, as compared to misuse of weak identifiers.

Our notion of identity verification deals with verifying that the identity attributes claimed by an individual are also owned by that individual. Identity verification is coupled with the concept of identity assurance. The notion of identity assurance deals with the confidence about the truth of the claims related with the identity of an individual. Successful identity verification with high assurance about an identifier claimed by an individual means that the identifier is considered valid and the verifier is confident that it is owned by that individual.

Strong and weak identity assurance exist regardless of the linkability of the identifier to the human identity of the actual individual. Additionally linkability among identifiers may exist with or without being bound (or linked) to the actual individual. Various cases exist which are summarized in Table 1.1 and further elaborated in the following example.

Table 1.1 Matrix of identity assurance and identity linkability combinations.

Identity Assurance \ Linkability	Strong	Weak
Strong	Case 1	Case 2
Weak	Case 3	Case 4

Example 1 Consider an individual whose real world name is Bob Smith who has a digital pseudonym $Homer07$. In a digital interaction when $Homer07$ claims to have a $SSN = 123456789$ and the verifier has strong assurance that the claim is correct (i.e. the SSN is valid and owned by the user $Homer07$) and linked to the real world individual Bob Smith, then this corresponds to Case 1 of Table 1.1.

Consider another scenario in which $Homer07$ claims to be have $citizenship = U.S.A.$ and the verifier does not know which real world individual does the claim belong to, but at

the same time, is confident that the claim is correct. Such scenario corresponds to Case 2. Notice that for a party to make a decision, such as in access control, linkability to a human identity of the actual individual is not always required.

Case 3 and Case 4 correspond to the situation in which the verifier is not confident that the claim is correct; the difference is that in Case 3 the verifier knows which is the real individual presenting the claim, whereas in Case 4 the verifier is not aware of who this individual is.

In our view, the Case 1 and Case 2 are more interesting as high assurance on the correctness of identifiers would also lead to trustworthiness and confidence on the use of such identifiers.

In addition to the traditional identifiers, there also exist *biometric* identifiers that are increasingly included as an integral part of individuals' identity. Biometric verification occurs when an individual presents a biometric sample, and possibly some additional identifying data such as a password, which is then compared with the stored sample for that individual. Biometric verification provides some inherent advantages as compared to other non-biometric identifiers because biometrics correspond to a direct evidence of the personal physical characteristics versus possession of secrets which can be potentially compromised. Moreover, most of the times biometric enrollment is executed in-person and in controlled environments making it reliable for subsequent use [4].

1.1.2 Identity Theft

The management of identity attributes raises a number of challenges caused by conflicting requirements. Although identity attributes need to be shared to speed up and facilitate authentication of an individual and access control, they also need to be protected as they may convey sensitive information about an individual and can be targets of attacks such as identity theft. By identity theft we mean the act of impersonating others' identities by presenting stolen identifiers or proofs of identities. More precisely, the act of identity theft occurs when an individual successfully uses an identity attribute or proof of an identity

which he/she does not own. Usually, identity theft in the digital world occurs to obtain credit or to perform other crimes, such as accessing classified records without having the appropriate authorization. People are increasingly concerned about identity theft as it is a serious economic crime. In 2005, the Consumer Sentinel, a Federal Trade Commission (FTC) complaint database, received over 685,000 consumer fraud and identity theft complaints [5]. As of July 2006, the U.S. Justice Department reported charging 432 individuals with aggravated identity theft for the 2006 calendar year [6]. U.S. President George W. Bush has called identity theft "One of the most harmful abuses of personal information" [7]. In fact, the average monetary loss per victim attributed to the crime of identity theft is more than the amount attributed to bank robbery [8]. There is also increased federal and state legislation regarding identity theft that has brought a heightened awareness to identity theft in general. For instance, the Identity Theft and Assumption Deterrence Act of 1998 makes identity theft a federal crime (18 U.S.C. § 1028 (2003)). The purpose of this statute is to criminalize the act of identity theft itself, before other crimes are committed. Several other regulations concerning protection of personal information are presented in Appendix A. Through attacks such as password cracking, pharming, phishing [9], and database attacks, malicious parties can collect sensitive identity attributes of (targeted) individuals and use them to impersonate these individuals or to sell the identity attributes. *Even though technical solutions are available that mitigate such attacks [10], a comprehensive approach to the problem of identity theft cannot rely solely on these techniques and must be able to offer protection from the threat of identity theft also when these solutions fail.* We provide an approach which offers protection in every stage of the identity lifecycle including the issuance, storage, usage, modification and revocation of identity. More specifically, we focus on strong identity verification, which is fundamental in preventing identity theft.

1.2 Evolution of Digital Identity Management Systems

One of the key reasons for the initial development of IdM systems was the proliferation of identity silos among various organizations and also within an organization. By identity

silo we mean an identity store containing individuals' identity data that is specific to an organization or application within an organization, which cannot be used by a different organization or other applications in the same organization. A new identity silo is created each time a different identity store is needed for a new application because existing identity stores cannot be reused because of interoperability problems. In most cases off-the-shelf identity systems such as Microsoft Active Directory [11] service may be used for the new applications, thus resulting in the creation of new identity silos. This silo-model still remains as the most predominant IdM system deployed in the current-day corporate world. This has made identity provisioning cumbersome for the end user and the IdM system restricted and inflexible.

Therefore the so-called *centralized model* such as Microsoft Passport [12] emerged, which examined a possible solution to avoid the redundancies and inconsistencies in the silo model. Under the centralized model the individuals single sign-on (SSO) such that they can authenticate once and gain access to the resources of multiple software systems. Here a central Identity Provider (IdP for brevity) became responsible for collecting and provisioning individuals' identity information. This approach has several drawbacks as the IdP is potentially the single point of failure and in several cases not trusted by all participating parties.

As a next step, the goal was to decentralize the responsibility of the IdP to multiple IdPs that can be selected by the end-users. Such systems are often coupled with the notion of *federation* [6, 13]. The goal of federation is to provide individuals with protected environments to share identities among organizations by managing individuals' identity attributes. Federations provide a controlled method by which federation members can provide more integrated and complete services to a qualified group of individuals. The members of a federation have trust relationships amongst themselves to share and use individuals' identity attributes.

Federations are usually composed of two main entities: IdPs that manage identities of individuals, and service providers (SPs) that offer services to registered individuals. In a typical federated IdM, the individual registers with his/her local IdP and is assigned a login

name. Based on this information a registered individual can submit additional attributes and corresponding attribute release policies that are stored at the IdP. From then on, the IdP is contacted whenever the individual interacts with any SP in the federation and additional identity information is needed. The IdP is then in charge of sending the SP the submitted attributes of the individual in accordance with the attribute release policies. In such *federated systems*, multiple IdPs are distributed and can store partial identity information of individuals, if required. Federations typically do not have the problem of single point of failure, but an IdP must be chosen that is also trusted by other entities. In most of these systems individuals have thus to depend on an online IdP to provide the required credentials and hence these systems are referred to as *provider centric* [6, 13, 14]. In some cases, such systems do not provide user control on his/her identity information, which is one of the key drawbacks of such systems.

As a result, an emerging paradigm in federated IdM systems is that of *user centricity*, where the idea is to provide the individual full control of transactions involving his/her identity information. This paradigm is embraced by multiple industry products and initiatives such as the Identity Mixer Project [15]. There are several terms closely associated with the concept of user centricity, including “user control,” “user consent,” and “user in the middle.” In our work, by user centric we mean the user has control on the use of his/her attributes. Having good user control also implies strong security properties such as non-repudiation and stealing prevention. Interestingly, the silo model may provide user control, however, as mentioned above, this is cumbersome for the user. Thus, the new federated IdM systems need to incorporate the advantages of previous approaches, for example SSO and decentralization of IdP and at the same time provide further user control on his/her identity information.

There are other diverse systems emerging which are often categorized as IdM systems and which do not fall in the above mentioned categories. One example is the form filling IdM that supports the user when filling forms by automatically inserting or suggesting input values. These systems provide useful, but limited functionalities. We however focus on the more comprehensive federated IdM systems that accomplish tasks such as provi-

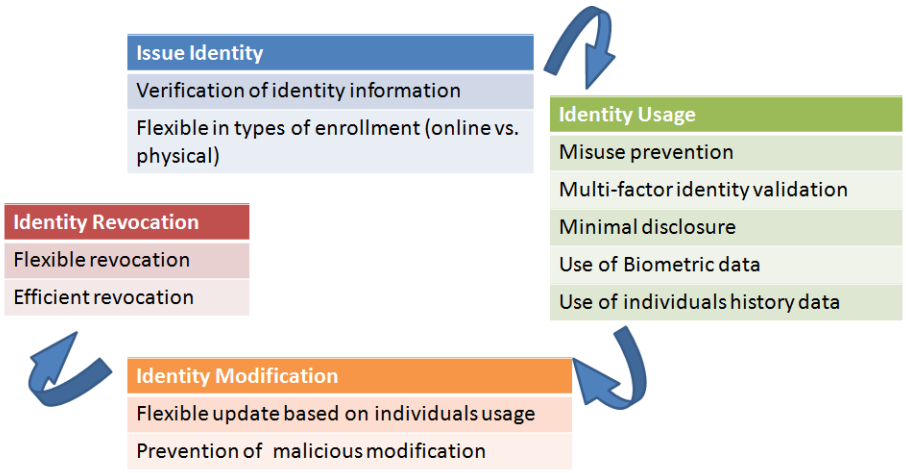


Fig. 1.1. Shortcomings of current federated IdM approaches in the identity lifecycle.

sioning, multi-factor authentication, access control, data protection, auditing and policy enforcement. It is useful for such upcoming IdM systems to make a conceptual distinction of functionalities related to the management of identity data and the use of such data. IdM systems are constantly evolving to meet the growing needs that integrate both the management and the use of identity information. In this thesis we focus on a specific part of such comprehensive systems that relates to identity verification and misuse prevention.

Shortcomings of current federated IdM Approaches. Several projects and initiatives are investigating digital IdM for federated systems [6, 13, 14, 16, 17]. Based on the simple identity life cycle illustrated in Figure 1.1, we identify some general shortcomings of current approaches that relate to both provider centric and user centric IdM systems.

Issue Identity. First, a limitation is that no information is provided about whether the strong and weak identifiers being enrolled and stored at the IdPs have been verified to be correct with respect to validity and ownership, and the strength of this verification. If an IdP

has such information then the SPs are in a position to make a more accurate judgment concerning the trustworthiness of such identity information.

Second, most IdM systems lack flexible enrollment mechanisms for the individuals who want to enroll in their systems. Enrollment can be in-person at a physical location of an IdP or online. Current systems however, do not provide for alternative mechanisms for individuals to enroll. Moreover, the types of identity attributes that can be enrolled in most systems are also restricted based on the nature of the IdP organization [6].

Identity Usage. A major drawback is that no specific techniques are provided to protect against the *misuse of identity attributes* stored at the IdPs and SPs. Even the notion of misuse of such attributes has not been thoroughly investigated yet. By misuse we refer to the case when dishonest individuals register fake attributes or impersonate other individuals of the federation, leading to the threat of identity theft.

To mitigate this threat, an upcoming trend is represented by *strong authentication*. Strong authentication often refers to systems that require multiple factors -possibly issued by different sources- to identify users when they access services and applications. However current approaches to strong authentication (such as those deployed by banks, enterprises, and governmental institutions) are neither flexible nor fine grained. In many cases strong authentication simply requires two forms of identity tokens, for example password with biometric. Through prior knowledge of these token requirements, an adversary can steal the required identity information to compromise such authentication [2, 18]. Moreover if the same tokens are repeatedly used for strong authentication at various SPs, then the chances of these tokens being compromised increase. Thus the implemented strong authentication [6] does not meet the stronger protection requirements of identities in a federation. Individuals should be able to choose any combination of identity attributes to perform strong authentication provided that the authentication policies defined by the verifying party are satisfied.

Another drawback, in the identity usage phase, is the inability of the individuals to *disclose minimal identity* data about themselves to the SPs and IdPs as per the need of the

service requested. There are several security and privacy concerns related to the extraneous identity information of the individuals that are stored at the SPs and IdPs. Moreover, such data may be aggregated or used in a manner that could potentially violate the privacy requirements of the individuals on their data.

Current approaches also do not address how *biometric data* can be used in their system; in that digital identities are defined by digital attributes and certificates. The use of biometrics as an integral part of individual identity is gaining importance. At the same time, because of the nature of the biometric data, it is not trivial to use such data in a way similar to the traditional attributes. It should be possible to use biometric data together with other identity attributes to provide protection against identity attribute misuse.

Another type of identity data that is not supported in current systems is the one related to individuals' histories of online activities. If this information can be verified and used for evaluating properties about an individual, for example reputation, then this information becomes a part of the individuals' identity. For example, consider a scenario where an individual frequently buys books from an online store. This *history based* information can be encoded as an identity attribute of that individual, which in turn can be used to evaluate the reputation of this individual as a buyer.

Identity Modification. Most approaches do not provide flexible mechanisms to update or modify enrolled identity attributes. As the information is shared within the federation, the updates performed on one system do not ensure consistency of the individuals' information within the federation. Additionally, these systems fail to prevent malicious updates by attackers that impersonate the honest individual.

Identity Revocation. Finally, current federated IdM systems lack practical and effective revocation mechanisms. To enable consistency and maintain correctness of an individual identity information revocation should be feasible. Revocation in provider centric systems, in which the IdP provides the required credential to the user each time, is relatively simple to solve. Such credentials are typically short term, and cannot be used without consulting the issuer again. If, however, the credentials are stored with the user, such as a long-term

credential issued by the appropriate authority, then building a revocation system becomes more challenging and critical.

1.3 Desired High Level Properties

Current identity management systems [6, 13, 14, 16, 17] do not leverage their underlying system architecture to develop techniques to protect individuals from identity theft. Dishonest individuals can register stolen attributes or impersonate other individuals of the IdM system. Protection from identity theft should be one of the main desiderata in all IdM solutions. Even if the identity attribute of an individual is stolen, the system should make it hard for an adversary to use it successfully. Verification of identity attributes is a key component of any solution to the problem of identity attribute misuse.
Other important requirements for a secure and privacy preserving identity system are as follows:

1. IdM systems main resource is represented by attributes, thus *security* of such information should always be guaranteed. Security includes a comprehensive set of properties, including integrity, confidentiality, revocability, and non-repudiation of ownership of identity attributes as described in Section 1.4.

2. Identity verification methods should preserve individuals' *privacy*, and enforce a "need to know principle" [19] when requiring identifiers. Privacy refers to the concept of giving an individual control over the release and use of his/her attributes. In this context *data minimization* is required, in that only the attributes actually required to access a service should be submitted to the SP. Data minimization can be achieved by a combination of appropriate policies, and data release mechanisms supporting selective release of information.

3. A federated identity system should ensure *consistency* of the identity data shared within the federation. Although validity of identifiers can only be verified by checking with actual identifier issuers, which could be outside the federation, the system

should be able to detect misuse of identity attributes based on the information available within the federation.

4. The verification methods should be *efficient* and require a limited number of message rounds between the SP and the individual. This would be one way to ensure usability of the system, as it is one of the main aims of federations.

5. The system should be able to support a variety of identity attributes, including *biometric* data and individuals usage *history* data referring to his/her online activity.

1.4 Main Techniques and Contributions

The goal of this dissertation is to formally define and analyze approaches to the problem of identity theft in a federated IdM system. Specifically we develop techniques for identity verification, satisfying comprehensive security and privacy properties derived from the high level requirements provided in the previous section. These techniques are built in the context of a framework, which we refer to as VeryIDX.

The underlying basic properties providing the security and privacy properties are illustrated in Figure 1.2 along with the key mechanisms used to achieve them. The properties are described briefly in Table 1.2, in the context of a typical IdM where there are mainly three kinds of entities – SPs, IdPs and individuals. SPs provide services to individuals, and IdPs issue certified attributes to individuals, store such attributes and provide them in a controlled fashion. By transaction in an IdM system, we mean a sequence of actions taken by an individual, IdP and SP, to provide the identity data the SP requires to provide requested services to the individual. The basic properties apply to a combination of 1) the entire IdM system, 2) transactions in the system, and 3) the identity information of the individuals involved. Though this classification is not exclusive, the semantics of the properties highlight which of the three entities they are relevant to. In what follows we describe the main techniques we use to achieve such properties. Thereafter we highlight the key contributions of the thesis.

Table 1.2 Basic properties achieving security and privacy properties.

Confidentiality deals with the protection of information from unauthorized disclosure. This property applies to identity information and transactions in the system. Identity information should only be accessible by the intended recipients.
Integrity requires data not to be altered in an unauthorized way.
Revocation of identity information is required to maintain the validity of the information in the system. It should ensure that once invalid information is recognized, it is not used for identity verification purposes.
Unlinkability of two or more users or transactions means that the attacker after having observed the transactions, should not gain additional information on linking those transactions. Unlinkability prevents (illegitimate) merging of user profiles by linking them.
User-choice means that the individual can choose between multiple IdPs and which attributes to release.
Verifiability means that the individual can verify that the IdP provides the correct identity data about the individual and according to the individuals intention. As such, an individual giving her consent about what data is revealed, for what purpose and to whom, means that the individual's view of the transaction corresponds to the actual transaction and that the individual agrees to the execution of the transaction.
Non-replay of the identity data provided in transactions prevents unauthorized parties from successfully using an individuals identity data to conduct new transactions. Non-replay is one prerequisite for obtaining the non-repudiation property.
Non-repudiation of transactions and identity data itself means that the sending of a non-repudiable identity data cannot be denied by its sender and the ownership of the identity data cannot be denied.
Stealing protection applied to identity data is concerning the issue of protecting against unauthorized entities illegitimately retrieving individual's data items. Stealing protection is required to achieve properties such as non-repudiation.
Selective release of identity information means that identity information can be released at a fine-granular level as controlled by the individual. In this way an individual can provide only the identity information that needs to be released for a service without having to release additional information.

1.4.1 Main Techniques

The identity verification solution is based on two key elements. The first element is the notion of *multi-factor verification of identity attributes,* which consists of verifying that

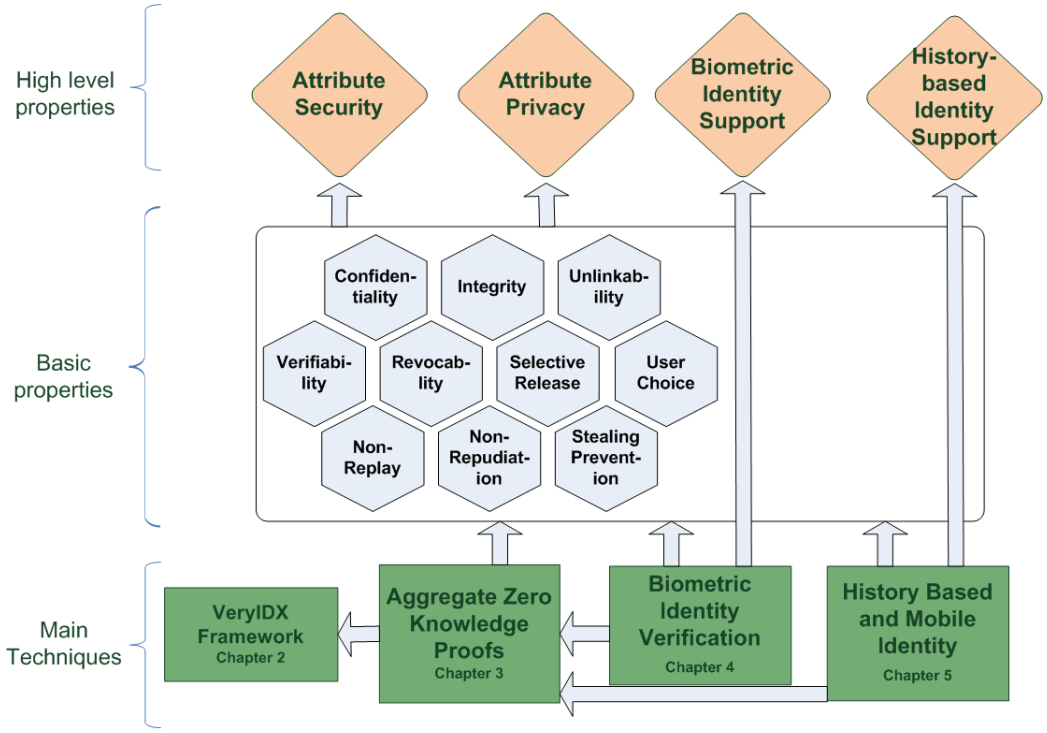


Fig. 1.2. Desired properties and main techniques.

an individual owns an identity attribute by requiring from this individual other associated proofs of identity, that is, of other strong identifiers. Note that we use the concept of *proof of identity*, which consists of a cryptographic token bound to an individual, versus the actual value of the individuals' attribute. A proof is created in such a way that only the individual to whom the proof is bound can properly use it. Proofs of identity attributes are built using Zero Knowledge Proof of Knowledge (ZKPK for brevity) [20, 21] techniques. The second key element is the notion of *identity assurance level*, that is, a level associated with identity attributes that indicates the degree of confidence that the federation has in a certain identity attribute. Thus, the level indicates how strong the verification is for a given identity attribute. Such level is important for SPs in the federation when making decisions about granting access to services or resources.

The multi-factor identity verification protocols we propose are supported by efficient cryptographic primitives. We have developed a mechanism to prove the knowledge of multiple strong identifiers stored as cryptographic commitments using new aggregated, ZKPK protocols. The commitments are signed by a special federation entity, referred to as *registrar*, and the corresponding signature can be verified in an aggregated fashion at the time of use. To achieve aggregate signature we develop techniques based on the approach originally proposed by Boneh *et al.* [22]. Boneh's signature techniques are not sufficient as they do not support the signature of cryptographic commitments that can be used later for ZKPK protocols. We therefore use Pedersen commitment and integrate it with Boneh's bilinear aggregate signature scheme to establish a new cryptographic primitive for aggregate proof of knowledge on those commitments.

We develop specific functions and protocols to prevent a malicious entity from secretly misusing identifiers belonging to other individuals. Different individuals may claim the possession of a same identifier, which actually has a unique owner. To address this problem, we use a mechanism based on distributed hash tables (DHT) [23, 24] that efficiently detect identifier duplicates in a federation. Duplicates of identifiers may be a signal of an identity theft attempt. The duplicates can be detected through the stored deterministic commitment of the strong identifiers in the DHT.

As the goal of our work is to provide a comprehensive approach to the problem of identity theft we explore the use of biometrics in the context of IdM systems. Today a large number of biometric devices and techniques are available and biometric-based solutions are increasingly being deployed [25, 26]. It is thus important that our framework be able to incorporate identifiers encoding information about physical features of individuals, in addition to "attributive" identifiers (such as SSN). The introduction of biometrics poses several non-trivial security challenges because of the inherent features of the biometric data itself. Biometric matching is probabilistic in nature, which implies that two samples from the same individual are never exactly the same. To preserve privacy and achieve interoperability between biometric identifiers and the other identifiers, we develop a biometric key generation algorithm. We build on mechanisms from image hashing [27] and data classifi-

cation techniques [28]. We use Singular Vector Decomposition (SVD) on biometric images to derive a hash vector representing the biometric. Biometric images of the same individual would result in 'similar' hash vectors. The similarity is evaluated using a Support Vector Machine (SVM) that classifies the hash vectors. We use the classification information to generate the final biometric-keys. Such keys are used to generate ZKPK similar to the other strong attributes of the individual.

The notion of *history based* identity attributes is motivated by the fact that such history can provide reliable information about the individual characteristics and behavior based on the online activities of the individual. We extend our approach to support history based identity information in the context of e-commerce transactions and history based trust management systems in which information about past transactions of the individual is used to make trust-based decisions concerning current transactions [29]. It is important that these decisions be based on reliable transaction history information and that misuse of this information be prevented. Our approach supports the management of history based identity attributes that are encoded as electronic receipts in the VeryIDX framework. We have devised a series of receipt protocols that use identity-based signatures, contract signing and certified email protocols, in addition to our ZKPK based identity verification protocols, to achieve specific privacy and security properties. Moreover we show how the receipts can be used in a mobile environment. The current convergence of telecommunication and computer network technologies is resulting in a broad range of personal devices and sensors. We use the specific design features and capabilities of the Near Field Communication (NFC) enabled cell phone devices while extending the receipt protocols for the mobile environment. NFC is a standards-based, short-range ($\sim$ 15 centimeters) wireless connectivity technology that enables two-way interactions among electronic devices [30].

In the next section we provide the main results obtained employing the above techniques.

1.4.2 Contributions

This thesis research provides the first robust and comprehensive solution to the problem of identity verification and theft prevention in a federated IdM environment. Our approach combines different novel techniques, each of which addresses a specific issue that arises when dealing with identity verification. These devised techniques are general and can thus be potentially applicable beyond the IdM framework. The key contribution are as follows:

1. A new cryptographic primitive referred to as *aggregate proof of knowledge* to achieve privacy preserving multi-factor verification used in the VeryIDX framework. This primitive uses aggregate signatures on commitments that are then used for aggregate zero-knowledge proof of knowledge (ZKPK) protocols. The resulting signatures are short and the zero-knowledge proofs are succinct and efficient. We prove the security of our scheme under the co-gap groups for Diffie Hellman (co-GDH) assumption for groups with bilinear maps. Our cryptographic scheme is better in terms of the performance, flexibility and storage requirements than existing efficient ZKPK techniques that may be used to prove, under zero-knowledge, the knowledge of multiple secrets.

2. Algorithms to reliably generate a cryptographic key from an individual's biometric image using SVD based hashing functions and SVM classification techniques. Our algorithms capture generic biometric features to ensure unique and repeatable biometric keys. We provide an empirical evaluation of the proposed techniques using 2569 images of 488 different individuals for three types of biometric images; namely fingerprint image, iris image and face image. Based on the biometric type and the classification models, as a result of the empirical evaluation we can generate keys ranging from 64 bits up to 214 bits.

   We use the generated keys to perform multi-factor identity verification. Several factors, including various traditional identity attributes, can thus be used in conjunction with one or more biometrics of the individual to provide strong identity verification. The algorithms ensure specific security and privacy properties related to the biometric data and the biometric verification. More specifically, we analyze several attack

scenarios including the case when all the data stored on the client machine is compromised, even in such a case the biometric key is not revealed. The privacy of the biometric is preserved, in that no information about the original biometric image is revealed from the biometric key.

3. A series of protocols for the establishment and management of individuals' transaction history based identity attributes using receipts from e-commerce transactions. These receipt protocols satisfy the security requirements related to the management of the electronic receipts, namely correctness, integrity, single submission, fairness and non-repudiation. All receipt protocols are privacy-preserving with respect to user consent and minimal disclosure of the receipt attributes.

   We show how the user's receipt protocols can be employed in the context of NFC cellular phones. In particular we provide techniques for selectively, yet securely, managing the identity information on such devices, and using them at physical locations of the SPs. This enables portability of identifiers, which are used for identity verification while ensuring the desired security and privacy properties hold.

1.4.3 Advantages of the Proposed Approach

Our approach has several advantages discussed below:

- Privacy of individuals is preserved, as minimal information is released, both in the registration and the usage phase. Individuals only register the identifiers they are willing to commit. At the time of usage, the actual values of identifiers are revealed only if required for obtaining the service. Additional proofs of identity can be provided by the individuals without revealing the actual values of identity attributes. The verification methods are efficient, because individuals can satisfy SPs multiple identifier verification requirements by disclosing a single piece of information. Because of the aggregate ZKPK protocol, efficiency is ensured even if proofs of multiple identifiers are required.

- The federation protocols are secure with respect to the basic security and privacy properties described in this section. Even if some information about individual identifiers is leaked to an adversary, the adversary is not able to use it for obtaining any service in the federation. The main effort required by an individual is when it first establishes identity proofs. Once this bootstrapping part is completed, the operations needed from the individual are minimal. The protocol proofs required for verification may be implemented without requiring any human intervention if the secrets are stored in tamper proof hardware.

- Our approach makes it possible to maintain consistency in a federation with respect to two well known invariants of individuals identifiers. First, strong identifiers are generally unique, unless proved otherwise by the owners. The second invariant is related to the fact that several strong identifiers of an individual have some common weak identifiers associated with them. The two invariants cover the common understanding of the notion of strong identifiers.

- Biometric identifiers are supported. The introduction of biometric verification into a framework for the verification of identity attributes is novel and will result in advances to the state-of-art with respect to the integration of cryptographic protocols and biometric data in IdM systems.

- History based identity attributes are supported. They provide a way to use individuals online activity to generate reliable identity information which can be managed and used as any other identity attributes to evaluate reputation and other trust relationship based related properties.

- The approach supports portable identifiers and their usage with mobile devices such as cellular phones. Several aspects relevant to such devices with respect to the security and resource usage are investigated.

1.5 Organization of the Dissertation

We start our discussion by presenting our framework for federated digital identity management system in Chapter 2. Chapter 3 is devoted to cryptographic protocols and mechanisms for supporting privacy preserving multi-factor identity verification. Here we also provide a detailed and formal analysis, demonstrating the security, efficiency and flexibility of our approach using the devised protocols. Chapter 4 provides our biometric key generation algorithms and detailed analysis of our methods. Chapter 5 describes our approach to history based identity attributes which includes several receipt protocols and how they can be used with mobile devices. Chapter 6 presents current state of the art from the literature of identity management systems, privacy enhancing systems and other technologies related to the techniques presented. Chapter 7 summarizes the thesis work and points out other applications of the proposed techniques as well as the future directions of this work.

2. A FRAMEWORK FOR FEDERATED IDENTITY MANAGEMENT SYSTEM

Developing a robust and comprehensive solution to the problem of identity verification satisfying manifold security and privacy requirements in an identity management system is a complex task. To address this task we have developed the VeryIDX framework that focuses on identity verification of individuals in a federated identity management system. This framework supports a step by step approach according to which an individual can first establish a digital identity followed by a secure and protected use of such identity.

Generally, federations rely on PKI for exchanging data among SPs, and for individuals to identify SPs [31]. However, PKI has experienced numerous implementation problems because of its technical complexities while using it in the context of identifying individuals [32]. It is also oriented towards unique identification of the individual, granted through Registration and Certification Authorities, which is not always suitable for individual privacy. Hence, the assumption of relying on PKI for all types of interaction involving individuals in the federation may not be feasible. We thus need articulated identity solutions supporting multiple complementary options for digital identity.

In this chapter we illustrate the VeryIDX framework for a federated system that assigns SSO IDs to individuals within the federation, and subsequently allows individuals to add other identity attributes with strong guarantees against identity theft. Our protocols do not rely on PKI for identity verification of an individual, so that one can use uncertified attributes and be eligible for services with low clearance. For cases in which certificates are required, we show how certificate issuers can leverage the SSO ID to issue certified attributes to individuals. We can employ PKI protocols if a PKI infrastructure is available for individuals.

The key idea behind multi-factor identity verification is to associate the different strong identifiers, possibly issued by different issuers, with each other and with the individuals'

SSO ID. At the time of use, the enrolled strong identifiers are acceptable only when the proof of ownership of one or more of the associated strong identifiers is provided. We refer to a set of such strong identifiers as *Secured from Identity Theft* (SIT for brevity). These SIT identifiers are associated with cryptographic commitments and can be used by individuals to generate proofs of identity without revealing the attributes in clear. In our approach, SIT identifiers (or attributes) are protected: if a user wants to use any of its SIT identifiers, it has to provide one or more associated SIT identifiers as proofs of identity. We show how with the help of cryptographic techniques presented in Chapter 3 we can preserve user privacy without jeopardizing security.

We note that strong cryptographic techniques built upon identifiers that have weak identity assurance are vulnerable to misuse. Therefore before presenting how SIT attributes are established and used, we elaborate on the techniques that are used to evaluate the identity assurance of the SIT attributes. Then we show how identity assurance techniques are used while managing SIT attributes. Revocation mechanisms of the SIT identity attributes are also developed. In particular, we illustrate how the revocation techniques can benefit from the underlying VeryIDX federation framework.

This chapter is organized as follows. In the next section we introduce basic components of the VeryIDX framework. Section 2.2 introduces the concept of SIT attributes and gives an overview of our approach to the problem of identity theft with illustrative examples. Section 2.3 provides details on the identity assurance of the SIT attributes, which is crucial for the correct usage of such attributes. Section 2.4 and Section 2.5 elaborate on the management and revocation of the SIT attributes. In Section 2.6 we provide a summary.

2.1 Basic Components

In this section we first introduce the notion of VeryIDX federation that will be used throughout the rest of the presentation. We then present preliminary concepts related to establishing individuals identity attributes.

2.1.1 VeryIdX Federation

The goal of federations is to provide users with protected environments to share identities by the proper management of identity attributes. Federations provide a controlled method by which federation members can provide more integrated and complete services to a qualified group of individuals. We employ a simple federation model that is however sufficient for developing our solutions for identity verification in federated IdM systems. Our approach to federations involves three types of entities: principals, service providers, and registrars. Formally, a VeryIDX federation is modeled as a tuple $\mathcal{F} = \langle \mathcal{P}, \mathcal{SP}, \mathcal{R} \rangle$, where:

- $\mathcal{P}$ is the set of **principals**. Principals are associated with a SSO ID; each SSO ID represents a principal of the federation. An individual can be associated with several principals in the federation.

- $\mathcal{SP}$ is the set of **service providers** (SP). The services offered by SPs are protected by a set of policies defining the requirements principals have to satisfy for their use. Such policies are referred to as service disclosure policies [33, 34].

- $\mathcal{R}$ is the set of **registrars** which establish and maintain *identity commitments* used to establish proof of knowledge of SIT identifiers. At least one registrar has to be in place to achieve multi-factor identity verification of principals.

The distinction among the above entities is logical and therefore the same federation host may provide the functions associated with several such entities.

We assume that public keys managed under public key infrastructure (PKI) and related standards are used for identifying SPs and registrars. The use of such infrastructure provides secure communications among entities. The use of PKI credentials for authentication between federation entities is employed in several current federation frameworks [6,13,35].

The identity information of the individuals stored at the registrar is signed by the registrar at the time of enrollment (see Section 2.4.1). For this purpose, the registrar has an additional public-private key pair that allows the registrar to aggregate signatures as pro-

posed by Boneh *et al.* [22]. In Chapter 3 we show how the aggregate signature of the registrar on principals multiple SIT strong identifiers can be efficiently verified.

2.1.2 Identity Attributes

Our notion of digital identity is defined in Chapter 1, Section 1.1. As mentioned digital identity refers to nyms and partial identities. A partial identity is a set of strong and weak identifiers (or identity attributes[1]) associated with an individual.

In our work, we employ SSO IDs (nyms) to uniquely identify a principal associated with an individual within a federation. Individuals who enroll with the registrar are identified by their chosen user name and the registrar name, separated by symbol $@$. That is, if the chosen name of an individual is $Alice$ and it enrolls with registrar $Reg1$, then the SSO ID would be $Alice@Reg1$. Note that other user naming conventions could be used here. The essential property is that a principal's SSO ID be unique within the federation.

Once the individual has successfully established a SSO ID in the federation, then this SSO ID is used to represent the corresponding principal. The principal can enroll different types of identity attributes associated with it. The SP services that a principal can be eligible for depend on the satisfaction of the identity verification and service policy enforced by the SP. Such policies specify the principal's identity attributes that are required for identity verification and to qualify for a particular service.

Identity attributes can be further distinguished into two types: 1) uncertified attributes, corresponding to voluntary information given by individuals; and 2) certified attributes, corresponding to attributes that have been verified and issued as signed digital certificates [36, 37] by trusted third parties. Using its SSO ID the principal can log on to different SPs and access the provided services. Here, different scenarios may arise. In case the principal does not have any certified attributes, it can access services for which only voluntary information needs to be provided.

[1]Note the word identifier is used synonymously with identity attribute.

For services requiring higher clearance and thus requiring certified information from the principal, the principal has to be issued the required digital certificates from third party trusted authorities called certificate authorities (CA's). There is an agreement among federation entities about which CA's are trusted. Trust on a CA may be based on an acceptable, well-defined procedure for the verification and certification of different attributes [38, 39] that is followed by the CA. Therefore the certificates issued by such CA's will be considered reliable within the federation.

There are different approaches that can be taken by a principal to obtain certified attributes or certificates. In the case when a principal does not possess any initial digital certificates, the certificate issuance may be performed at a physical location of a CA, so the principal can obtain digital certificates based on the principal's real world credentials. For example consider a case when a principal who has a SSO ID $Alice@Reg1$, does not possess any digital certificate and needs to obtain a digital certificate certifying that *Alice_SSN* is the social security number (SSN) belonging to the principal $Alice@Reg1$. In this case, the individual corresponding to the principal has to show the physical credentials verifying the ownership of *Alice_SSN* and the $Alice@Reg1$ SSO ID, to authorized personnel at the physical CA office. If the verification is successful, the principal receives a certified attribute or certificate that asserts that *Alice_SSN* belongs to SSO ID $Alice@Reg1$.

A second approach can be used when a principal either already has some digital certificates, or the claimed information can be verified by accessing some reliable online databases. Additional certificates can be issued based on this information. We assume that policies to issue such certificates are in place at the CA's. CA's can also issue a special type of certificate that attests the ownership of other certificates. These certificates help associate user certified identity attributes from different certificates. An example to illustrate this is as follows:

Example 2 Figure 2.1 demonstrates an example for the issuance of a certificate that denotes credential ownership. As shown, principal Alice has two certificates. The first certificate is issued by a registrar and states that $Alice@Reg1$ has SSN *Alice_SSN*. The second certificate is from a trusted CA and states that *Alice_SSN* has a *Low_Income _Status*, (LIS)

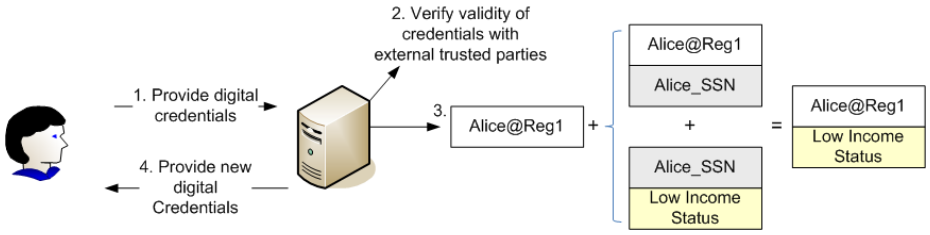


Fig. 2.1. Example showing ownership of credential certificate issuance.

for brevity. Alice wants to get a certificate stating that she (represented by her SSO ID) has a LIS to be used within the federation without revealing her SSN. Alice can obtain such a certificate by submitting to a trusted registrar the LIS certificate and the certificate associating her SSN with her SSO ID. In return she obtains the final certificate associating $Alice@Reg1$ with LIS. Here, the actual disclosure of the SSN may not be required, but the value only needs to be the same for the two certificates.

Following is an additional example of certificate issuance based on CA policies.

Example 3 When a principal requires the issuance of new certificates, it must prove possession of pre-requisite certificates according to the certificate provisioning policies of registrars. If $Alice@Reg1$ has the certificate associating the SSO ID with her SSN, she may be eligible to obtain a *Trusted-User* certificate. In this case the certificate provisioning policy of the registrar may require Alice to be uniquely identified so she can be held accountable.

As shown by the example above, when a principal requires the issuance of new certificates, it must prove possession of pre-requisite certificates according to the certificate issuance policies at the CA's. In the example unique identification using a strong identifier is used here for accountability purposes. However, this condition is not always needed for accountability if pseudonymous systems [15, 40–42] or using alternate techniques based on weak identifiers [2] are used.

Certificate management techniques to achieve the above tasks have been explored extensively [36, 43, 44]. We focus on the use of such certified identifiers for SIT identifiers in the rest of the chapter.

2.2 Identifiers Secured from Identity Theft

Identity theft occurs when a malicious individual uses an individual's personal information such as the user's name, Social Security number (SSN), credit card number (CCN) or other identifying information, without its permission. In this section we offer our approach to prevent identity theft in a federation. As mentioned earlier in this chapter, we refer to the strong identifiers secured from identity theft as SIT strong identifiers (or simply SIT identifiers/-attributes). The identity verification using SIT attributes is based on a well known technique called zero knowledge proof of knowledge (ZKPKs for brevity) [41, 45]. ZKPKs allow an individual to prove the possession of a private secret without releasing it. SIT attributes are associated with information theoretically secure Pedersens commitments [46] which are used to provide proof of knowledge of the corresponding strong identifier.

These SIT attributes can correspond to both certified or uncertified strong identifiers, although we focus primarily on certified strong identifiers. Registrars are assumed to be semi-honest[2] for the principals' attributes they keep track of. A principal can register its SIT attributes with *any* registrar in the federation by first engaging in a bootstrapping SSO ID enrollment procedure. Once the initial registration is completed, a set of SIT attributes are associated with the principal's SSO ID and with each other. These attributes are used together with ordinary data to protect from identity theft. Here, and throughout the chapter, by protection against identity theft we mean the inability to use a SIT attribute without the proof of additional identity information. To protect against identity theft it is important that an adversary be prevented from registering attributes of other principals as its own SIT attributes; therefore our security model includes identity assurance mechanisms elaborated

[2] According to the accepted definition of semi-honest entities, we assume registrars will follow the protocol but may also want to learn more information than they are supposed to.

in Section 2.3. We provide our approach to achieve the above functionality in the following section.

2.2.1 Main Approach

The key components of our solution can be summarized as follows:

1. Whenever a principal P presents a SIT strong identifier to a SP in the federation, the SP requires additional proofs of identity according to its local verification policies; each SP may have different verification policies. The submission of these additional proofs of identity by P and the corresponding verification by the SP is executed through the use of our new aggregated ZKPK protocols (See Chapter 3). With our aggregated proofs the principal can prove knowledge of any combination of several SIT strong identifiers efficiently. As the actual values of the identifiers are not revealed to the SP this approach preserves the privacy of the principals. We show that a malicious principal cannot provide these proofs of identity, unless the values of all the relevant SIT strong identifiers and the secrets associated with them are compromised.

2. Each SIT strong identifier used by a principal P in a federation, either for direct use or only for identify proof, must be registered with a registrar that, upon registration, provides P with a signature on the commitment of the identifier. Identifiers can be registered at different times and also when the party is performing an interaction with a SP. The management of the registered SIT strong identifiers is based on *identity records* (IdR) created for each registering party. IdRs are elaborated further in this section.

3. To avoid that a malicious principal registers with a federation a strong certified identifier owned by another individual, a duplicate detection protocol is run upon registration to determine whether the same strong identifier has already been registered by a different party. Duplicate detection requires that prior to registration, the principal

contacts the original issuer of the strong identifier, which generates a unique number which is associated with the commitment of the strong identifier in a certificate signed by this issuer. This is achieved through the blind signature protocols, where the blind value of a strong identifier corresponds to the semantically secure commitment of this identifier, which is subsequently used in the registration. At the time of registration, duplicates are detected based on the use of distributed hash tables (DHT).

The Identity Record (IdR) for a principal is established at the time of registration and collects the commitments corresponding to the SIT identifiers. Each commitment in the IdR is signed by the registrar. The IdR also collects other parameters along with a SSO ID and relevant weak identifiers. The IdR is used to evaluate the *identity assurance* of the enrolled commitments. The IdR is updated with information about the registered individuals as they progressively use the registered strong identifiers at the various SPs. Individuals can register new strong identifiers *on-the-fly* at the registrar where their IdR is stored. Registrars use the IdRs to determine the correctness of the registered strong identifiers and to detect theft attempts. Each time a commitment is registered, the registrar ensures that the values are consistent with the information present locally at the registrar and globally within the federation.

The content of the IdR has to be available when the individual requests a service from a SP for enabling verification of the commitments. Availability can be ensured in two ways. One way is to let the individual indicate its registrar[3], so that the SP can directly retrieve the required content of its IdR. This approach requires the registrar to be online. The other option is to encode the IdR content in a certificate, and have the registrar sign each tuple. This structure can be stored at the individual and then submitted along with the service request. To avoid the usage of stale certificates, the certificates can be short-term and re-issued to the principal on a regular basis.

[3]In general several registrars may be part of a federation.

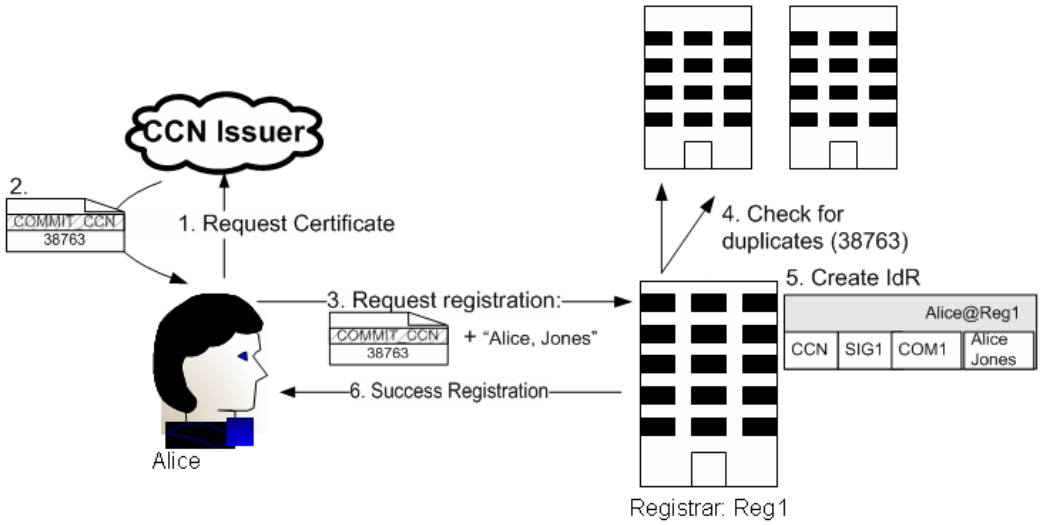


Fig. 2.2. Example 4 illustrating the main approach (Steps 1 to 6).

2.2.2 Examples

We now provide a case scenario showing an application of our solution for SIT attributes. The steps are also illustrated in Figures 2.2 and 2.3.

Example 4 Consider Alice, an individual who wants to join federation E-Mall. E-Mall offers a safe environment for online business, comparison shopping, web hosting, domain registration, banner advertising, website advertising and so forth. Alice first establishes a SSO ID and password with a registrar $Reg1$. Before registering and using the SIT identifiers, Alice contacts the issuer of the strong identifier to get a certificate that contains the commitment of the strong identifier, created by Alice, and a unique number associated with the strong identifier that is generated by the issuer. For example, in step 1 in Figure 2.2, she contacts her bank, who issued her a credit card number (CCN), to get a certificate that uses the blinded or committed value of the CCN and associates it with a unique number corresponding to it. Then in step 3, she registers her CCN with registrar $Reg1$ to be safely used within the federation.

$Reg1$ upon receiving the CCN first verifies that the submitted CCN has not been already registered by some other individual in federation E-Mall as depicted in step 4. If this condition holds, Alice and $Reg1$ execute the registration protocol to sign the desired commitment for the CCN, which will be used subsequently when this CCN is used as a proof of identity. This information is encoded in Alice's Identity Record, which can also be encoded in a certificate signed by the registrar if desired.

Alice, as a member of the federation can now proceed, as in step 7 in Figure 2.3, with the request of a service from an E-Mall store, say SP-Shop. According to the SP-Shop's policy, in the next step, this store requires Alice's CCN along with a different form of identity verification from Alice for authentication. SP-Shop thus challenges Alice's SSN. Note that SP-Shop has no actual interest in knowing the value of the SSN; but it wants to be assured that (i) Alice knows the SSN of the owner of the CCN and that (ii) they both refer to Alice (as identified by her first and last name). Because SSN is a strong identifier, Alice wants to protect it from identity theft. As such Alice submits an additional request, in step 11, to $Reg1$ for registering her SSN. $Reg1$ does the necessary checks to see if the information provided in clear (e.g. first name and last name) is consistent with Alice's record, and that no one else has registered the same SSN. If these checks are performed successfully, then $Reg1$ updates the IdR of Alice with the new SSN commitment. At step 14, a corresponding certificate storing the IdR could be re-issued to Alice, and the previous one discarded.

Alice, in step 15, is now in the position to send the updated certificate and to be able to construct the aggregate proof of knowledge of the required identifiers for the SP-shop to resume her original request. SP-Shop verifies the ownership of the certificate with the help of the proof, followed by validating the CCN itself. In step 17, CCN validation information is sent to $Reg1$ if the transaction is successful. This way, $Reg1$ now knows that the CCN registered initially is valid and can update this information in Alice's IdR. Alice can thus complete the transaction.

Our approach supports the strong verification of identity attributes, which is a component of comprehensive solutions against identity theft. Here, a basic requirement is proving

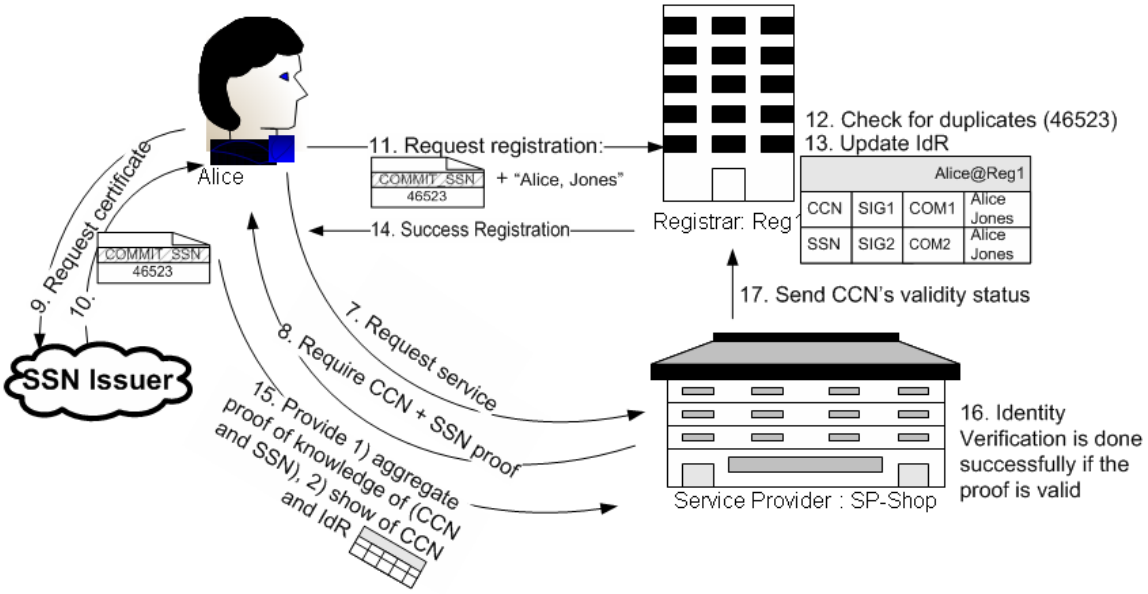


Fig. 2.3. Example 4 illustrating the main approach (Steps 7 to 17).

the knowledge of the various committed values and this is achieved through our multiple proof of knowledge with aggregate signatures cryptographic technique. If all but one secret of the multiple factors are stolen by an adversary the proof cannot be constructed. Moreover, this combination of knowledge of strong identifiers is determined at the time of identity verification and is computed efficiently, adding the required flexibility and usability in the system.

We now extend the above example, to the case when a malicious party tries to use a stolen identifier and show how our technique protects against such attempts.

Example 5 Consider a malicious user Mallory, who has managed to steal Alice's CCN after Alice has registered it. Mallory's main goal is to be able to use Alice's CCN to shop at SP-Shop. When SP-Shop challenges Mallory to provide the SSN, corresponding to this CCN, Mallory could try one of two possibilities. First, she could attempt to use the public

IdR of Alice and try to construct the proof of knowledge of the registered commitments. Mallory is however not able to construct such proof, as our aggregate proof of knowledge is possible only if the prover knows the actual values of the committed strong identifiers. Moreover, Mallory does not gain any advantage, even if she manages to learn the value of the SSN and not the random secrets associated with the commitments. The second possibility Mallory has is to register the CCN of Alice, with her own SSN. However, when Mallory sends a request to register the CCN, because of our duplicate detection mechanism, the registrars can verify that the same strong identifier has already been registered and aborts the registration. Thus Mallory cannot use Alice's CCN, once Alice has registered it with the federation.

Thus we see how our system mitigates attacks from malicious users to impersonate and misuse the compromised SIT strong identifiers of the registered users in the federation.

2.3 Identity Assurance in VeryIDX

The notion of identity assurance deals with the confidence about the truth of the claims related with the identity of an individual. Weak identity assurance may increase the risk of identity theft, as provenance and authenticity of the identity data are not certain. Hence, strong cryptographic techniques built upon identifiers that have weak identity assurance are vulnerable to misuse. Strong identity assurance is thus a crucial requirement for any identity management system. Our approach to identity assurance relies on the use of multiple proofs of identity that are stored in the IdR. The IdR is a fundamental notion of our approach in that it actually provides a digital representation of user identities. In this section we define all the components of IdRs and various notions of assurance related to such records. We describe the requirements and mechanisms to evaluate and maintain assurance about IdRs in subsection 2.4.

We now elaborate on the notion of Identity Record and the required security checks for ensuring identifier consistency within the federation.

2.3.1 Identity Record and Assurance Levels

As we mentioned, each principal P in a federation has associated one or more IdRs, each recorded at some registrar in the federation. Each IdR in turn consists of several *identity tuples*, denoted as τ's. Each identity tuple is associated with one SIT identifier and records all information required for the verification of this identifier at the time of use[4]. In particular, each SIT identifier m is associated with two other types of identifiers: a semantically secure commitment, and a unique identifier associated with m, denoted as M and $\widehat{M}$, respectively. M is generated by the principal while interacting with the issuer and submitted for registration. M is then signed by the registrar upon registration of m.

The signature on M is denoted by σ and is part of the identity tuple associated with m. M is computed as $g^m h^r$, where g and h are generators in group G of prime order q. The commitment is created in a form that is used in the ZKPK presented in Chapter 3. G and q are public parameters of the registrar and r is chosen randomly from $\mathbb{Z}_q$[5]. To allow principals using different IdRs at different registrars, the public parameters are the same for all registrars in the federation. $\widehat{M}$ is a unique identifier which is assumed to be generated by the issuer, at the time of issuance of the certified strong identifier m. In the following, we refer to $\widehat{M}$ as the *identifier binder*. The issuer encodes the unique identifier $\widehat{M}$ along with the commitment M in a certificate, denoted as Υ. M is encoded in a hidden fashion. Υ is signed by employing blind signature techniques [41, 42, 47, 48] on the original strong identifier m. It is issued to the principal, and it is crucial in determining re-registration or duplicates, as illustrated in Section 2.3.3. Note that $\widehat{M}$ is conceptually tied to m and this relation can be proven based on Υ. Therefore $\widehat{M}$ itself is not to be stored in the identity tuple.

m is also tied to a set of weak identifiers, denoted by $\{w_1, \ldots w_k\}$. For example, assume 404033004379 to be a credit card number and 'Alice' and 'Smith' be the first and last name of an individual. Here, 404033004379 is the strong identifier value, while 'Alice'

[4]Note that τ can possibly contain null values for information that is not known.

[5]These public parameters may need to be known by the issuer while creating the unique identifier. More details of the cryptographic commitments and mechanisms are subject of Chapter 3.

and ‘Smith’ are the associated weak identifiers. The IdR is also associated with some other public parameters required for the cryptographic protocols, as detailed in Chapter 3 Section 3.3.

All SIT strong identifier commitments and weak identifiers are tagged with an identifier descriptor tag and two types of assurance, namely *validity assurance* and *ownership assurance*. Validity assurance corresponds to the confidence about the validity of the identifier based on the verification performed at the identifier’s issuer. As such, it refers to the correctness of identifiers with respect to the real world information sources and the issuers of the identifiers, which can possibly be external to the federation. For example an issuer (say $MasterCard$) can verify if a credit card number it issued is valid. Ownership assurance corresponds to the confidence about the claim that the principal presenting a given identifier is its true owner.

We introduce four levels of assurance: absolute assurance, tagged as ‘A’, corresponding to the absolute certainty about the claim; reasonable assurance, tagged as ‘B’, corresponding to case when one or more assertions from trusted parties exist regarding the certainty of the claim; unknown assurance, tagged as ‘U’, when there is no information to assert the certainty of the claim; and false assurance, tagged as ‘F’, denoting that the claim is incorrect.

We assume that absolute validity (label ‘A’ of the validity-assure) of a given strong identifier can only be determined by authorities that have issued the strong identifiers. Because such authorities may not always be part of a given federation, we assume that the federation is allowed to verify validity of identifiers with such authorities according to existing agreements. Note however that our approach also supports the case when such verifications are not possible. We mark as ‘B’ the validity assurance of a strong identifier, the validity of which has been asserted by a principal and this principal has other several identifiers with ‘A’ category validity-assurance. If no entity other than the principal supports the validity of the strong identifier, the identifier is marked with unknown assurance ‘U’. Identifiers might be immediately validated by contacting the corresponding issuers, either upon registration, or they might be validated later on when actually used by the principal.

Alice@Registrar1 | *PARAMS*

Strong IdTag	Signature [σ]	Commit-ment [M]	valid-assure	owner-assure	WeakID (list)
CCN	74387264 87979976 66876989	3298397 9798749 3827983	A	B	Value | tag | valid | own Alice | fname | A | B Mars | lname | A | B
SSN	88874724 72323098 40923610	3987239 8747973 8294991	U	A	Value | tag | valid | own Alice | fname | A | A 12442 | zip | A | A

Fig. 2.4. Simplified graphical representation of an Identity Record.

The latter corresponds to the concept of *lazy validation*. Precisely, strong identifiers can be validated following either a *pull* or a *push* approach. Under the *pull mode*, the registrar determines the validity status before the value of commitment is signed and used in the federation. Thus, the registrar checks the validity immediately and the validity assurance is set to 'A'. Alternatively, the push mode implements the lazy validation. The validity is verified by any SP in the federation receiving the actual value of the strong identifier. The SP contacts the corresponding issuer, and sends the validation result to the registrar storing the corresponding IdR so that the IdR can be updated accordingly.

The notation adopted to represent the various IdR elements is as follows:

IdR = $\{\{\tau_i\}$, cryptographic parameters $\}$

τ_i = $[(\sigma_i, M_i, tag, validity\text{-}assure, ownership\text{-}assure), \{W_{i_j}\}]$

W_{i_j} = $(w_{i_j}, tag, validity\text{-}assure, ownership\text{-}assure)$

The resultant IdR is used by the principal to perform multi-factor identity verification using the enrolled strong identifiers. The strength of the verification is based on verification policies of the verifiers that may use the ownership and validity assurance of the enrolled SIT identifiers. We now provide an illustrative example of an IdR in Figure 2.4 and explain it with the following scenario.

Example 6 Consider a principal known as Alice@Registrar1 enrolled with registrar referred to as Registrar1. She has registered two strong identifiers: a CCN and a SSN. The signatures and commitments of each are computed and stored. The CCN is validated in the *pull* mode by Registrar1, therefore its $validity\text{-}assure$ is 'A'. The CCN is enrolled online and the $ownership\text{-}assure$ is 'B'. The SSN is instead enrolled through a face-to-face interaction. Its $validity\text{-}assure$ is set to 'U' as the value has not been confirmed by any entity other than Alice, while the $ownership - assure$ is set to 'A'.

2.3.2 Ownership and Consistency of Identity records

To effectively protect identity of registered principals, all the identity records in the federation should collect only those identifiers, the ownership of which has been assured. An implied aspect to take into account deals with identifier's consistency across the federation. Before formally defining these two key notions we need to clarify the concept of *proof of knowledge* of a strong identifier of value m. We say that principal P can provide a proof of knowledge of m if it provides a verifiable cryptographic token used in a ZKPK protocol (see Chapter 3, Section 3.3) that asserts that P knows: (1) the actual value of m, and (2) cryptographic secret(s) associated with the corresponding semantically secure commitment M. This is denoted as $M \rhd P$ (read "M belongs to P").

The conditions according to which a principal P can prove ownership of a given IdR are dictated by a policy π. Such policy specifies which of the committed strong identifiers in a given IdR need to be proven to ensure ownership of the whole record. Policy π can either be specified separately by the various registrars or it can be globally defined as a part of the federation agreement policies. Ownership of IdR by P proven according to policy π is denoted by $IdR_P \blacktriangleright_\pi P$ and is formally defined as follows.

In the definition and throughout the chapter we adopt the dot notation to denote an element in a given object. That is, a tuple τ' appearing in IdR is denoted by $IdR.\tau'$, and the tag descriptor of the strong identifier in τ' as $\tau'.tag$.

Definition 2.3.1 (Ownership of Identity Record) *Let π be a policy and let ψ be the set of tags to be verified according to π. A principal P registered at registrar R can prove ownership of an identity record IdR_P if, for each $t \in \psi$, $\tau \in IdR_P$ exists, such that $\tau.tag = t$ and the following conditions hold: (i) $M \rhd P$; (ii) P can provide proof that signature $\tau.\sigma$ is valid ; (iii) $\tau.validity\text{-}assure = A$; (iv) $\tau.ownerhip\text{-}assure = A$ or B.*

The definition states that ownership assurance of an identity record is the result of the ownership assurance of each tuple referred in the policy. That is, for each such tuple, it is required that the signature on the strong identifier be verified, the validity assurance of the strong identifier is set to 'A' and the ownership-assurance is set to level 'A' or 'B'.

Example 7 With reference to Example 6 (see also Figure 2.4), Alice's ownership assurance on the first tuple in the IdR corresponding to the CCN, is 'A' because the validity assurance of the CCN is 'A' and ownership assurance of the CCN is 'B'. However, her ownership assurance on the second tuple corresponding to SSN, is 'U' because the validity assurance of the SSN is 'U'. Thus, by taking a conservative approach, the ownership assurance of Alice's IdR is 'U'.

Notice that identity assurance is a broader concept than only ownership and validity assurance. Identity assurance is also related to the *consistency* of the IdR. Consistency of IdR is both a local and a global concept. Local consistency deals with the information recorded by a specific IdR. To be consistent, the collected strong and weak identifiers should qualify an individual with no evident errors. That is, no conflicting attribute values should be collected in the same IdR. For instance, if the weak identifier "age" appears in different weak identifiers, it should have the same value. Global consistency requires that no strong identifier be associated with multiple principals, as they are typically unique, unless some specific conditions, detailed below, hold.
We formalize the concept of consistency in the following definition.

Definition 2.3.2 (Consistency of Identity Records) *Let $\mathcal{F} = \langle \mathcal{P}, \mathcal{SP}, \mathcal{R} \rangle$ be a federation. Let P be a principal in $\mathcal{P}$, enrolled at registrar $R \in \mathcal{R}$. Let IdR_P be an identity record of P. We say that:*

1. Idr_P *is locally consistent with respect to* R *if* $\forall \tau_i \in Idr_P, \nexists \tau_i.W_{ih}, \tau_j.W_{jk}, j \neq i | tag_{W_{ih}} = tag_{W_{jk}}$ *and* $\tau_i.W_{ih} \neq \tau_j.W_{jk}$.

2. Idr_P *is* globally consistent *with respect to* $\mathcal{F}$ *if one of the following conditions holds:*

   (a) $\nexists P_j \in \mathcal{P}, P \neq P_j | \widehat{M_i} = \widehat{M_j}$ *and* $\tau_j \in Idr_{P_j}$ *and* $Idr_{P_j} \blacktriangleright_\pi P_j$

   (b) if $\exists P_j \in \mathcal{P}, P \neq P_j | \widehat{M_i} = \widehat{M_j}$ *and* $\tau_j \in Idr_{P_j}$*, then* $Idr_{P_j} \blacktriangleright_\pi P$.

3. *We say that* IdR_P *is* consistent *if it is* locally consistent *with respect to* R *and* globally consistent *with respect to* $\mathcal{F}$.

Local consistency checks are executed to verify that there are no weak identifiers in the same IdR with the same descriptor tags and different values. For example the value of the weak identifier tagged by *firstname* should be the same in all strong identifiers in which it appears. With respect to global consistency, the first condition requires that no duplicates of strong identifiers exist in a federation; in most cases, a strong identifier is unique to an individual and a duplicate may represent an inconsistency. However, as stated by the second condition, if a duplicate is detected in another identity record, then the principal should be able to prove ownership of this identity record. Therefore, our approach also allows multiple principals to commit the same value for strong identifiers, under the condition that ownership of the duplicate strong identifier can be proven. For instance, we let two principals share a same credit card, if both can prove the ownership of the corresponding IdR.

A summary of identity assurance types and levels is presented in Table 2.1. This illustration is used in the rest of the dissertation. However the framework can be applied to other assurance levels and the associated policies defining the requirements for each level. Identity assurance can be achieved by enforcing specific checks at registrars.

2.3.3 Functions Enforcing Identity Assurance

We have designed a set of functions implementing the controls needed to verify identity assurance. These functions are summarized in Table 2.2 and details are provided as follows.

Table 2.1 Summary of identity assurance types and levels.

Assurance Type	**'A'**	**'B'**	**'U'**	**'F'**
Validity-assure	Validated by original issuer	Offline checks using algorithms with known correctness criteria of the identifier	No validation done	Proven false
Ownership-assure	In-person registration	Digital introduction (§ 2.4)	No ownership proof	Proven false

LocalConsistency(τ, IdR, P) . This function checks whether the weak identifiers $W = (w_j, tag, validity - assure, ownership - assure)$ appearing on the input identity tuple τ are locally consistent. That is, if the same tag descriptors are present in any other identity tuple τ' of the same IdR, the value of the corresponding tags should be equal to w'_j. For example, if a credit card number (CCN) was committed along with weak identifiers $firstname$ and $lastname$, when a new identifier, say SSN, is committed the values of the weak identifiers $firstname$ and $lastname$ associated with the SSN should be identical to that for CCN[6].

FederationDuplicateDetection(τ, IdR,P) This function checks whether duplicate values of m exist in the federation, where m is a strong identifier appearing in tuple τ belonging to *P*. Duplicate detection can be achieved if the strong identifiers m are enrolled using certificates Υ which are uniquely identified, as introduced in Section 2.3.1. The blinded value in Υ is the semantically secure commitment M.

At the time of registration the individual provides the uniqueness token (e.g. $\langle M_i, \widehat{M_i} \rangle$) and the commitment M_i being registered. The registrar checks if a duplicate of $\widehat{M_i}$ exists. Duplicate detection is based on use of a Distributed Hash Table (DHT) [23]. This table keeps track of the strong identifier tokens that have validity status equal

[6]We assume the same tag names corresponding to the same semantic.

to ‘A’. The DHT is maintained by the registrars of the federation. The entries in the DHT are tuples of the form $\langle \widehat{M}, P, R \rangle$, where P denotes the principal, and R the identifier of the registrar storing the principal’s IdR. Duplicate detection is actually executed by running algorithm $lookup$ on the DHT. Because the tables are distributed, the duplicate lookup is efficient in that it does not require an exhaustive search. If a duplicate is found the algorithm returns false and further actions are taken to detect whether a misuse has occurred.

For completeness we briefly describe the DHT in the following. A DHT is not centrally stored in that it partitions a key space among *n* servers. The keys are mapped uniformly to the registrars. Such an approach enables horizontal partitioning of hash tables to distribute the execution of identity verification operations and data storage across the various registrars. Assuming a secure hash function is used [49], this horizontal partitioning strategy evenly distributes load and data across registrars. Each registrar has a partial list of where data is stored in the system. A lookup algorithm is used to locate data given the key for that data.

ExternalValidation(τ, IdR,P) This function validates the strong identifier m appearing in τ, by contacting the issuer authority, which provides validity assurance. If the issuer successfully validates m, the associated $validity - assure$ value is set to ‘A’. It is important that the weak identifiers used for the external validation correspond to the ones enrolled in τ.

External validation can be initiated according to the push or pull strategy. In the push mode, any SP in the federation receiving the actual value of the strong identifier m consults the issuer. The SP sends the validation result to the registrar storing the corresponding IdR so that the IdR can be updated accordingly. Under the pull mode, the registrar needs to determine the validity status before the value of commitment M is signed and used in the federation. In this case the principal encrypts its strong identifiers with the issuer’s public key which we assume to be available and sends it to the registrar. The registrar then appends the strong identifier binder $\widehat{M}$ to it and

Table 2.2 Identity assurance functions.

Function	Description
LocalConsistency(τ, IdR, P)	To check whether weak identifiers in τ are locally consistent.
FederationDuplicate Detection(τ, IdR,P)	To check whether duplicate values of the strong identifier commitment exists.
ExternalValidation(τ, IdR,P)	To validate the strong identifier m appearing in τ, by contacting the issuer authority.

sends it to the issuer for verification. Details of this procedure are given in Protocol 4b in Chapter 3. Once the verification from the issuer is completed, the registrar computes a signature σ on the commitment M and adds it to τ.

2.4 Management of SIT Identity Records

The management of identity in our approach is characterized by three main phases: enrollment, during which individuals register identifiers with the federation; usage of identifiers, requiring the verification of identity information; and update of identifiers, allowing individuals to modify their IdR. In what follows we discuss such phases in more detail.

2.4.1 Enrollment

Individuals are required to submit strong identifier commitments to enroll in the federation, according to the policy of the registrar. These strong identifiers are issued by various issuers, prior to the enrollment. As our approach is based on multi-factor verification of identity, we assume that a minimum number of identifiers is needed to actively participate in the federation. The exact type and number of identifiers to register is part of the registrar policy and is assumed to be publicly available from the registrar. For example, a registrar may require that a principal submits at least three strong identifiers for enrolling in the fed-

eration. As a registrar is not considered completely trustworthy, the values of the strong identifiers are not to be released in clear. The main goal of registration is thus to store unique and hidden SIT attributes to such semi-honest registrars.

When a principal P enrolls at registrar *R* a set of commitments $M_1, \ldots M_k$, corresponding to strong identifiers $m_1, \ldots m_k$ an IdR (say IdR_P) at R is created. The following steps are then executed for each m_i ($1 \leq i \leq k$) of the submitted strong identifiers:

1. The ExternalValidation (τ, IdR,P) function is executed, to assign the $validity$-$assure$ value of the strong identifier.

2. If $i \neq 1$ and thus other tuples have been inserted, the LocalConsistency (τ_{new}, IdR_P,P) function is executed to confirm local consistency of the identity tuples in IdR_P, with respect to the weak identifiers.

3. The FederationDuplicateDetection (τ_{new}, IdR_P,P) function is executed to check for duplicates of the various $\widehat{M_1}, \ldots \widehat{M_k}$.

4. The $ownership$-$assure$ level is determined based on the following three cases. If P is performing a face-to-face registration, then $ownership$-$assure$ = 'A', else if P's ownership claim is asserted by another principal who is trusted, then the $ownership$-$assure$ = 'B', and finally if no assurance is given, $ownership$-$assure$ = 'U'.

5. M_{new} is then signed by R to generate the signature σ_{new}.

6. Finally τ_{new} is added to the IdR. In this case the validation with the issuer has been performed, and the result of the $validity - assure$ is set according to the returned value.

When executing the FederationDuplicateDetection function if duplicates are found, the principal is asked to prove ownership of the IdR conveying the duplicate. If the principal is unable to provide proof of the ownership, then the enrollment is aborted. Once this check is completed the newly created IdR is consistent according to Definition 2.3.2. At the time of enrollment, ownership of the claimed strong identifiers also needs to be ensured.

Individuals can enroll either through a face-to-face registration process or online.

Physical Registration

Face-to-face registration is executed when an individual physically enrolls at a specific registrar office by showing credentials proving its identity and hence proving the ownership of the claimed identifiers. For example the individual can go to the physical registrar location where it shows its SSN card. A trusted official in the registrar confirms the validity of the physical card and supervises the enrollment procedure ensuring that the correct SSN number is entered into the system and thus stored as a commitment. Here the individual trusts the registrar's system not to store extraneous information other than the commitments needed for the enrollment. As per the example policy followed in the dissertation, face-to-face registration results in an $ownership\text{-}assure$ level equal to 'A'.

Online Registration

An alternative approach is online registration. Online registration of strong attributes is challenging in the absence of principals' public keys. To protect an individual's privacy, we require that strong identifiers of a principal are never given in clear, not even to the registrar storing this information, when such values are not needed to qualify for a specific service. This requirement adds a level of complexity to the registration procedure: the registrar cannot guarantee that the information registered is correct or owned by the principal enrolling it. We therefore base the online registration on the concept of *digital introduction* by strongly identified principals. The principal acting as a *grantor* for the enrolling principal needs to have a valid IdR (say $IdR_{grantor}$) and $IdR_{grantor} \blacktriangleright_{\rho} grantor$ where ρ is the identity verification policy of the registrar. The *grantor* essentially asserts that the enrolling principal actually possesses the strong identifiers the commitments of which are presented to the registrar. Such an assurance is based on the level of trust of *grantor*, the $ownership\text{-}assure$ of this type of registration has a level equal to 'B'. Online registration will thus require the individuals to present one assertion from at least one grantor in addition to the minimum number of strong identifier commitments.

2.4.2 Update

The IdR may have to be updated for 1) adding strong identifier commitments, 2) revoking strong identifier commitments and 3) changing the $validity - assure$ status of the strong identifier commitments in the IdR.

When a principal P requires adding a strong identifier commitment M_{new} to its IdR (say IdR_P) at registrar R, it presents an identity tuple τ_{new}, collecting M_{new} and a set of weak identifiers W_{new}; the following steps are then executed:

1. P proves ownership of IdR_P based on the policy of R, denoted asπ_R. Hence $IdR_P \blacktriangleright_{\pi_R} P$.

2. The LocalConsistency(τ_{new}, IdR,P) function is executed to confirm local consistency of the new identity tuple with respect to the weak identifiers.

3. The FederationDuplicateDetection(τ_{new}, IdR,P) function is executed to check for duplicates of $\widehat{M_{new}}$. If a duplicate is found at another IdR, say IdR_{dup}, then P has to prove $IdR_{dup} \blacktriangleright P$.

4. The $ownership$-$assure$ level is determined based on the following three cases. If P is performing a face-to-face update, then $ownership$-$assure$ = 'A', else if P uses digital introduction $ownership$-$assure$ = 'B', and finally if no assurance is given, $ownership$-$assure$ = 'U'.

5. M_{new} is then signed by R to generate the signature σ_{new}.

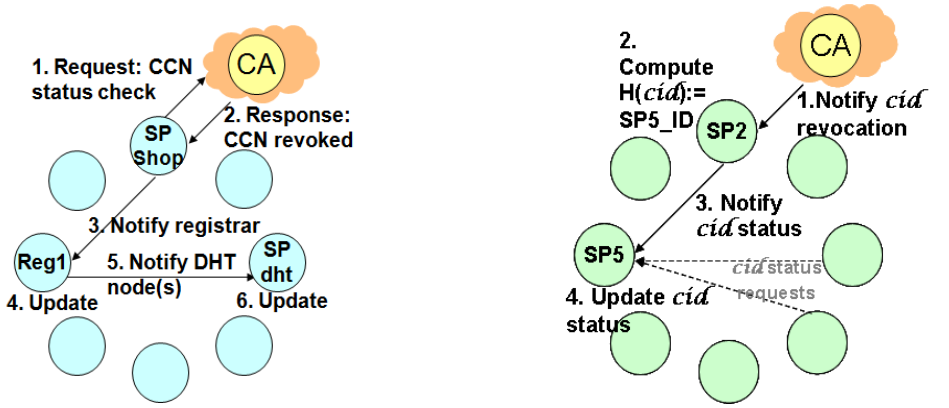
6. Finally τ_{new} is added to the IdR with the $validity - assure$ of each identifier set as unknown.

Revocation of a strong identifier commitment is executed by changing the $validity - assure$ to 'F'. The level of validity assurance is changed when the value returned from the execution of ExternalValidation function is different from that already stored in the IdR. A detailed approach on how to revoke SIT attributes is given in Section 2.5.

2.4.3 Usage

Any combination of the signed values of the commitments can be required for identity verification purposes by a SP. The content of the IdR must thus be available when the principal requests service from a SP. Availability can be ensured according to two strategies. One strategy is to let principal P indicate its registrar, so that SP can directly retrieve the required content of P's IdR. This requires the registrar to be online. Another option is to let P store the content of the IdR in a - per tuple - signed structure. Table 3.1 in Chapter 3 indicates which identity assurance functions and cryptographic protocols are used in the different stages.

2.5 Revocation of SIT Identifiers



(a) Steps for certified SIT attribute revocation in a *push mode*.

(b) New DHT for SIT attribute revocation in a *pull mode*.

Fig. 2.5. Revocation of SIT Identifiers.

Digital attributes state certain well-defined properties about the principals to which they refer. Such attributes may be indefinitely valid, or may be valid for a given time interval. Also, attributes may be revoked if some external events compromise their validity. *Revoca-*

tion thus refers to the *undo* of the claim associated with an attribute. Events that may cause attribute revocation include [50]:

- Compromise of the owner key: The owner key linked to this certificate has been compromised.
- Compromise of the issuer key: The issuer key used to generate this certificate has been compromised.
- Changes in the affiliation of the owner: The identification details of the certificate are no longer valid.
- Obsolescence of the certificate: The certificate is superseded by another certificate.
- Termination of the certificate: The certificate has reached the end of its validity period and has not been renewed.

2.5.1 Preliminary Notions Concerning Revocation

Most of the work in the area of revocation has focused on the revocation of certified attributes or public key certificates [51]. A widely used standard for defining these digital certificates is the X.509 [50] format. The two most widely-used schemes for managing X.509 certificate revocations are Certification Revocation Lists (CRL's) [50] and Online Certificate Status Protocol (OCSP) [52]. Both provide risk analysis based on certificate usage and efficient notification about the validity status of the certificate. CRL is essentially a list of certificate serial numbers that have been revoked and are therefore no longer valid. The CRL is always issued by the CA which issued the corresponding certificates. A PKI-enabled application consults this CRL to verify the validity of the certificate prior to its use. Because of its centralized nature, CRL is not scalable because of the bandwidth required to communicate with all its clients. OCSP supersedes CRL's by providing efficient notification through the use of a distributed protocol. A typical OCSP defines a request-response protocol between OCSP client and an OCSP responder. The OCSP responder is a trusted entity that informs the requester about the validity information of the

certificates. The OCSP responder can contact various backends including CRL's to retrieve the revocation information. One issue with OCSP is that the requester must know which OCSP is responder for a query. This information is typically specified in the Authority Info Access (AIA) extension of a certificate [52]. One limitation regarding the use of the AIA extension is that it may be difficult to deploy because of the increased certificate size. Moreover one has to ensure that there are no compatibility issues because of the different versions or types of certificates using the AIA extension. The problem of finding OCSP responders can be elegantly solved in a federated environment with the help of DHTs and with an optional use of the AIA extension. Another more important limitation of OCSP, and any current revocation mechanism, is that it cannot be used for uncertified attributes. This is because there is no assigned CA that can revoke such attributes. We leverage the federation architecture and follow a policy based approach for uncertified attributes. We assume that the revocation status is essentially provided by either the uncertified attribute's owner, or it is the result of the feedback of other federation entities.

A detailed description of the revocation mechanisms for the various types of attributes is given in the next subsection. We show how the underlying collaborative environment of a federation provides opportunities for efficient solutions to the problem of attribute revocation.

2.5.2 Revocation Techniques

The SIT attributes introduced are useful only if they can be verified and revoked reliably when necessary. It is required that revocation techniques be able to provide efficient notification to the potential consumer of the revoked attributes and prevent their subsequent usage. *When* and *how* should a SIT attribute be revoked depends on the type of the SIT attribute. As elaborated earlier, there are two types of SIT attributes, namely certified and uncertified, which require different approaches to revocation. The adopted approaches are described as follows:

Certified SIT Attributes. Certified SIT attributes should be revoked when the original issuer of the certificate (external CA, or an internal federation registrar) disqualifies that certificate. This corresponds to the credential revocation criteria already well investigated [50, 53].

Referring to Example 4, consider the case when $SP - Shop$ checks for the validity of Alice's CCN with the appropriate external CA and is notified that this CCN certificate is revoked. As a consequence of this notification, revocation steps have to be taken to update the information in the federation as shown in Figure 2.5(a). At step 3 $SP - Shop$ sends a signed revocation message to $Reg1$ with the SSO ID $Alice@Reg1$ and the tag CCN_{tag}. Based on this $Reg1$ can retrieve Alice's IdR (see Figure 2.4). Now based on the revocation policy of the registrar, $Reg1$ can accept the registration and either remove the row corresponding to CCN_{tag} or add an additional column to record the status information as *revoked*. In both the cases the *commitment* that is requisite for establishing proof of ownership of the corresponding SIT attribute is removed. As SIT attribute cannot be used without a valid commitment as shown in Chapter 3, subsequent usage of the revoked SIT attribute is prevented. This is the IdR update corresponding to step 4 in Figure 2.5(a).

In addition to updating the attribute information at the local registrar the DHT node (as introduced in Section 2.3.3, function FederationDuplicateDetection) saving the attribute commitment also has to be updated. This is because duplicates should be detected only for valid and unique identifiers. If an identifier has been revoked the federation policy may allow the re-registration of a revoked attribute or not. Therefore in step 5 $Reg1$ sends a revocation message to SP_{dht} that is the DHT node saving CCN's deterministic commitment. Depending on the revocation policy, SP_{dht} can either simply delete this information from its hash table or it can add the revocation information in the value corresponding to the commitment key. The update of the DHT node(s) completes the revocation process.

Uncertified SIT Attributes. Uncertified attributes correspond to voluntary claims of principals that do not have to be signed or verified by any trusted authority. Here the user itself is the issuer of the SIT attribute. Therefore the trust in the claim is often considered uncritical. However serious security problems, such as spam, phishing and pharming [10, 54–56] attacks arise from the incorrect usage of the uncertified attributes. It should thus be possible to revoke the usage of uncertified attributes. Revocation of uncertified attributes has not been explored.

Determining when uncertified attributes should be revoked is more complex than in the case of certified attributes because there is no entity who can assert accurately the validity of such attributes. However we assume that if a number of distinct revocation assertions are received for a certain attribute then the attribute is to be revoked. The number of accumulated assertions should be greater than a certain threshold, determined by the federation security policy. For example consider the case that user Alice subscribes her claimed email address $phony@myemail.com$ to SP_1, SP_2 and SP_3. Eventually because of bounced emails each of the SP concludes that the email is invalid. To revoke the attribute they separately send revocation requests to the designated registrar $Reg1$. $Reg1$ saves these requests in an additional column of Alice's IdR. If the federation accepts a threshold of three then when the third such request is received the $Reg1$ revokes this attribute. The revocation steps thenceforth follow steps 4 to 6 of the protocol for the revocation of certified attributes (see Figure 2.5(a)). Further usage of possibly incorrect uncertified attributes is thus prevented. A similar revocation procedure can be adopted if a principal who has performed multi-factor identity verification requests revocation of its attribute.

The notification mechanism described above for certified SIT attributes corresponds to a *pull mode* for revocation notification. This is a request reply mechanism where the reply is valid when it is from an authorized CA. In the case of OCSP, revocation information can also be from an authorized OCSP responder [52]. As highlighted earlier one problem is that the requester SP should know which OCSP responder it should contact to get the revocation information. One approach to address this issue is to define a *push mode*

revocation notification where the main CA pushes the revocation notification to the federation SPs when a revocation event occurs. Here the SPs themselves play the role of OCSP responders and the revocation information requests can be satisfied within the federation. To support such a solution we deploy an additional DHT (referred to as revoke-DHT for clarity) with the SPs as the distributed nodes. The revoke-DHT key in this case is the certificate ID itself. An external CA has to notify any one of the SPs in the federation. As an example in Figure 2.5(b) SP_2 is notified about the certificate identified by certificate ID cid. SP_2 computes the revoke-DHT hash $H(cid)$ to identify the DHT node (SP_5 in this case) where this cid should be stored. Subsequently it sends the revocation information to SP_5. Henceforth any SP that needs revocation information about certificate cid can directly access SP_5 by computing the same hash, thus identifying the DHT node responsible for providing cid's revocation information. This addresses the problem of identifying the OCSP responders outside the federation.

Note the hash values can be pre-computed and stored in the AIA extension field called the `accessLocation` [52]. This parameter essentially stores the location of the OCSP responder. AIA extension configuration is useful, but it has to be done carefully as improper use of certificate extensions has led to severe deployment problems [57]. Therefore instead of adding this information into the certificate we can leverage the knowledge of the revoke-DHTs hash function to calculate the responder at runtime. This information can also be cached locally in the system. In this way we provide two alternative methods that can be used to implement OCSP for certified SIT attributes.

2.6 Summary

In this chapter we introduced VeryIDX, which is an extensible framework for identity verification. VeryIDX employs a flexible and privacy-preserving approach that allows a user to establish basic identifiers and then proceed to establish other complex SIT identity attributes that are protected from identity misuse. Using example policies, we elaborate on specific identity assurance techniques that are critical while dealing with the SIT at-

tributes. Finally we show how such SIT attributes are enrolled, managed, used and revoked in VeryIDX.

Having the VeryIDX framework helps overcome the difficulties in understanding of how systems and protocols satisfying the desired set of security and privacy properties (See Figure 1.2) can be used within a complex IdM system. A comprehensive set of cryptographic tools, protocols and mechanisms presented in the rest of the dissertation are based on those specified assumptions, and serve as specifications for future development of such systems. More specifically, the cryptographic functions related to aggregate ZKPK presented in Chapter 3 are used to mathematically validate the resultant security and privacy properties regarding the multi-factor proofs within the VeryIDX system against the original requirements detailed in Chapter 1. The biometric and history based identifiers presented in Chapters 4 and 5 respectively, not only ensure the security and privacy of the respective identifiers, but also are used as strong identifiers in the cryptographic protocols. Figure 1.2 illustrates how the various conceptual components are interconnected within the VeryIDX framework.

There are specific assumptions to consider while employing the VeryIDX framework in any given IdM system. First, the existence of at least one registrar is assumed. This registrar is semi-trusted, in that the registrar follows the protocols for the management of SIT attributes, but it may maliciously try to retrieve or use the enrolled strong identifier values. The registrars do not need to be online during the multi-factor verification if the IdRs are stored as signed certificates at the principals. For global consistency checks performing duplicate detection and revocation protocols using DHTs, it is assumed that the registrars cooperate to exchange messages as defined in the protocols.

Second, we assume that the SPs and principals are untrusted and may try to misuse identity attributes. For the principals to use the VeryIDX multi-factor verification protocols it is assumed that they enroll the required strong identifiers with the registrars. It is also required that the principals secure and manage the secrets corresponding to their enrolled SIT attributes.

One important assumption during the decision process related to identity verification and assurance evaluation is the existence of policies associated with them. For multi-factor verification, the verifier's policies determine which factors are required from the principal. The strength of the verification would depend on how these verification policies are defined and enforced in the IdM system.

Policies for assurance level evaluation determine the conditions required for an identifier to achieve a certain level of assurance. We have provided representative examples of such policies that are used in the rest of the dissertation. However the usage of the IdRs is not limited to the provided example policies. This is because the design of IdR separates how the commitments are created and the policies used to evaluate the assurance on the various commitments. The commitments are created in a manner so that they can be used for mutli-factor proofs employing ZKPK protocols provided in Chapter 3. These commitments are not influenced by the policies that are used to evaluate ownership and validity assurance. Other policies can be applied to evaluate the assurance levels and recorded in the IdR. For example, there could be fine-grained policies to evaluate the resultant identity assurance. Those policies could use aspects such as identity provenance and trustworthiness of the software and hardware platforms used for identity management in the evaluation. Other work related to levels of assurance and metrics [58, 59] may also be used while evaluating the resultant identity assurance.

The VeryIDX logical framework provides a way to understand the key concepts related to providing privacy-preserving multi-factor identity verification. The set of operational scenarios using example policies cover the various possible cases that may be taken as a starting point to explore future possibilities.

3. MULTI-FACTOR IDENTITY VERIFICATION USING AGGREGATE PROOF OF KNOWLEDGE

The multi-factor identity verification requirement as highlighted in Chapter 2 is supported by protocols and use of new cryptographic primitives proposed in this chapter. Multi-factor identity verification consists of verifying the ownership of multiple strong identifiers of an individual. Note that by multiple factors we mean that the different strong identifiers come from different issuers, so they constitute independent forms of identity verification.

Our cryptographic protocols implement a mechanism to prove the knowledge of multiple strong identifiers stored as cryptographic commitments using aggregated zero-knowledge proofs. The commitments are signed by a special federation entity, referred to as *registrar* (See Chapter 2 Section 2.1.1), and the corresponding signature can be verified in an aggregated fashion at the time of use. To achieve aggregate signature we develop techniques based on the approach originally proposed by Boneh *et al.* [22].

Zero-knowledge proof of knowledge (ZKPK) is extensively used for identity protection [21, 60]. Our scheme enhances such protocols by the use of multi-factor proof. Although a single ZKPK has been proven to be sufficiently efficient [41], multi-factor proofs cannot maintain the same performance if a large number of proofs is considered. To address this issue we develop aggregated ZKPK and reduce the proofs of several factors that would require several ZKPKs to one that uses only one ZKPK. In our protocols, users always need to compute a small constant number of exponentiations, while the verifier's computation of exponentials is dramatically reduced, which makes our protocols highly suited for lightweight devices.

A key advantage of our protocols is that they are flexible with respect to which commitments are aggregated. That is, any combination of commitments (among the ones available for a given user) can be aggregated for computing the signature at runtime. This approach allows different SPs in the federation to challenge the knowledge of different combinations

of the committed identifiers. This is a substantial improvement with respect to existing approaches [21], which require the possible combinations of strong identifiers to be predefined or stored for computation of multi-factor proofs. Thus, under such protocols the space required is exponential with respect to the number of committed values. Our protocols instead require storing only the committed values and signatures.

Another main advantage of our solution is that, from an architectural point of view, it requires only minimal extensions. Besides the conventional set of IdPs and SPs composing a federation, our approach only requires adding some registrars. As mentioned in Chapter 2 the task of registrars is to enable users to register their strong identifiers without having to reveal their actual values. Registration in our approach means that users can establish cryptographic tokens called commitments, which can be used subsequently for establishing *proof of knowledge* for the corresponding strong identifiers. Registrars are small and modular software components that can be easily added to the architectures of current IdM systems. Our solution is also succinct and flexible, as we let the interacting entities exchanging only the information actually needed for the specific interaction; no extra information needs to be exchanged. Our protocols greatly reduce the amount of information revealed to the SP for verification of identifiers. We even provide a protocol that allows one to verify the signature of a commitment without knowing the value of the commitment itself. This property greatly enhances privacy while still assuring integrity and validity of committed data. Moreover the ZKPK commitments used are semantically secure requiring the enrollment of a random secret along with the strong identifier. The use of this technique ensures that even if an adversary learns the values of the strong identifiers, that is, steals identity information, it cannot wrongly present itself as the owner of this information.

The chapter is organized as follows. In Section 3.1 we provide the the preliminary concepts related to cryptographic building blocks used in our protocols. In Section 3.2 we provide the formal definitions for the aggregate ZKPK primitive. In Section 3.3 we present the cryptographic scheme for the aggregate proof of knowledge protocols. This is followed by a detailed analysis of security, efficiency and system security in Section 3.4. In Section 3.5 we provide a summary.

3.1 Preliminary Concepts

Following are the preliminary concepts regarding commitments, aggregate signatures and zero-knowledge proofs, and the corresponding protocol notation.

Pedersen commitments: Let g and h be generators of group G of prime order q. A value m is committed by choosing r randomly from $\mathbb{Z}_q$ and giving commitment $C = g^m h^r$ [46]. Commitment C is opened (or revealed) by disclosing m and r, and the opening is verified by checking that C is indeed equal to $g^m h^r$. A prover can prove by using zero-knowledge proof that it knows how to open such commitment without revealing either m or r.

Bilinear maps: For a security parameter k, let q be a prime of length k, and G_1, G_2, G_T be groups of order q. Suppose $g_1 \in G_1,\ g_2 \in G_2$ to be generators. Function $e : G_1 \times G_2 \rightarrow G_T$ is a bilinear mapping if it satisfies the following properties:

1. Bilinear: for all $u \in G_1, v \in G_2$ and $a, b \in \mathbb{Z},\ e(u^a, v^b) = e(u, v)^{ab}$.

2. Non-degenerate: $e(g_1, g_2) \neq 1 \in G_T$.

3. There exists a computable isomorphism ψ from G_2 to G_1, with $\psi(g_2) = g_1$.

Bilinear aggregate signatures: The aggregate signature concept has been proposed by Boneh *et al.* [22] based on the notion of bilinear maps. We refer to such signature scheme as BGLS. Informally, aggregate signatures are signatures that allow multiple signatures to be aggregated into one signature that is verifiable with respect the public keys of the signers and the signed messages. The BGLS scheme consists of five algorithms: $KeyGen, Sign, Verify, Aggregate$ and $AggVer$. Any principal P uses $KeyGen$ to generate the private and public key pair (χ, v) such that $v = g_2^{\chi}$, where $g_2 \in G_2$, χ is the private key, and v is the public key.

The $Sign$ algorithm computes the signature on input message m_i. Its main step is the mapping of m_i into G_1 by a mapping $h : \{0,1\}^* \rightarrow G_1$. The output message $\sigma_i = h(m_i)^{\chi} \in G_1$ is the signature for m_i.

The $Aggregate$ algorithm aggregates the signatures $\sigma_1, \sigma_2, \cdots, \sigma_t$ for t different messages $m_1, m_2, \cdots, m_t$ into one signature $\sigma = \prod_{i=1}^{t} \sigma_i$.

The $AggVer$ algorithm verifies a signature and works like the *Aggregate* signature algorithm. For a set $m_1, m_2, \cdots, m_t$ of different messages, and public keys $v_1, v_2, \cdots, v_t$ and a signature σ, the verifier checks if $e(\sigma, g_2) = \prod_i e(h_i, v)$, where $h_i = h(m_i)$ and e is the bilinear mapping.

Zero-knowledge proof of knowledge: In our approach we use the techniques by Camenisch and Stadler in [61] for the various ZKPK of discrete logarithms and proofs of the validity of statements about discrete logarithms. We also conform to the same notation as [61]. For instance to denote the ZKPK of values α and β such that $y = g^{\alpha}h^{\beta}$ holds, and $u \leq \alpha \leq v$, we use the following notation:

$$PK\{(\alpha, \beta) : y = g^{\alpha}h^{\beta} \wedge (u \leq \alpha \leq v)\}$$

The convention is that Greek letters denote quantities the knowledge of which is being proved, whereas all the other parameters are sent to the verifier. Using this notation, the proof protocol is described by pointing out its goal while hiding all details.

3.2 Definitions

In this section we provide the definitions for the aggregated ZKPK. We first review the concept of proof of knowledge and then define the new notion of aggregate ZKPK.

3.2.1 Zero knowledge Proof of Knowledge

An interactive proof system of knowledge for a certain relation $\mathcal{R}$ is a pair of algorithms, a prover P and a verifier V, the latter running in polynomial time. Informally, a proof of knowledge is an interactive proof in which the prover succeeds in 'convincing' a verifier that it knows something. What it means for a prover to 'know something' (say an element y) is defined in terms of computation, in that, a prover 'knows y', if on an input string x, it

can compute a relation $\mathcal{R}$ that depends on x and y without revealing y in clear.
More precisely, let $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ be a binary relation, then the witness set for any input element $x \in \{0,1\}^*$ is $R(x) := \{y \mid (x,y) \in \mathcal{R}\}$ and elements in $R(x)$ are witnesses of x. The language defined by $\mathcal{R}$ is $L_{\mathcal{R}} := \{x \mid R(x) \neq \emptyset\}$ is in NP and x is an element of the language. A relation $\mathcal{R}$ is a polynomially bounded relation if $\mathcal{R}$ satisfies that there is a polynomial $p(\cdot)$ such that $|y| \leq p(|x|)$ for all $(x,y) \in \mathcal{R}$.

In our approach, we consider only polynomial time computable and polynomial bounded binary relations. Therefore we let $\mathcal{R}$ be a polynomial time computable and polynomial bounded relation and P and V be PPT (Probabilistic polynomial time) interactive Turing Machines (ITM's). As mentioned P is prover and V is verifier in the interactive proof system. For any common input x and auxiliary input y to P, let $(P,V)(x,y)$ be the output of the verifier V in the execution of protocol $(P(x,y),V(x))$. $(P,V)(x,y) \in \{0,1\}$ and $(P,V)(x,y) = 1$ if and only if V accepts the proof. The system of ZKPK with knowledge error ϵ is given as follows:

Definition 3.2.1 ***(Zero-knowledge proof of knowledge, ZKPK)*** *The protocol (P,V) is a proof of knowledge for $\mathcal{R}$ if the following properties hold:*

- Completeness. *For all $(x,y) \in \mathcal{R}$, we have Pr$[(P,V)(x,y) = 1] = 1$.*
- Validity. *For every PPT ITM P', if Pr$[(P',V)(x,y) = 1] = p > \epsilon$, then there exists a PPT resetting ITM E such that Pr$[E^{P'}(x) = y \wedge (x,y) \in \mathcal{R}] = p - \epsilon$.*
  Here the resetting ITM E executes interactively with P' by (if necessary) resetting the random string of P' to reconstruct the desired knowledge.
- Zero-knowledgeness. *For any PPT ITM V^*, there exists a PPT ITM simulator M^* such that the following two ensembles are indistinguishable.*
  - $\{(P,V^*)(x)\}_{x \in L_{\mathcal{R}}}$
  - $\{M^*(x)\}_{x \in L_{\mathcal{R}}}$

The *completeness* property states that if the input x is in the language $L_{\mathcal{R}}$, then V always accepts the common input x after interacting with P whose auxiliary input is y.The *validity*

property requires that the success probability of a knowledge extractor E in extracting the witness, given oracle access to a possibly malicious prover P', must be at least as high as the success probability of the prover P' in convincing the verifier V about the proof of knowledge. This property guarantees that no malicious prover that does not know the witness can succeed in convincing the verifier. Finally the *zero-knowledge* property is formalized by saying that there exists an efficient algorithm (simulator M^*) that generates a pair of numbers that have distribution indistinguishable from the reference string x and the proof in a real execution of the proof system.

3.2.2 Aggregate Zero Knowledge Proof of Knowledge

Following from the concepts above, consider a series relations $\mathcal{R}, \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_m$, and the corresponding languages $L, L_1, L_2, \ldots, L_m$. Let H be a function from $L_1 \times L_2 \times \cdots \times L_m$ to L. Informally, aggregated ZKPK is to prove the knowledge of a lists of elements $x_1, x_2, \ldots, x_m$ by one proof of knowledge for an aggregated element $H(x_1, \ldots, x_m)$. After the successful execution of the ZKPK for $H(x_1, \ldots, x_m)$, the verifier will be convinced that prover indeed possesses the knowledge of witnesses $y_1, y_2, \ldots, y_m$ corresponding to $x_1, x_2, \ldots, x_m$ respectively. We provide a formal definition as follows.

Definition 3.2.2 ***(Aggregate zero knowledge proof of knowledge, AgZKPK)*** *For binary relations $\mathcal{R}, \mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_m$, and languages $L, L_1, L_2, \ldots, L_m$, a efficiently sample family of functions $\mathcal{H} = \{H \mid H : L_1 \times L_2 \times \cdots \times L_m \rightarrow L\}$, the interactive proof system (P, V) for R is an aggregated zero knowledge proof if the following hold: For any tuple $\overline{x} = (x_1, x_2, \ldots, x_m)$, there is a $H \in \mathcal{H}$ such that $x = H(\overline{x})$ with $(x, y) \in \mathcal{R}, (x_i, y_i) \in \mathcal{R}_i$, $\overline{y} = (y_1, y_2, \ldots, y_m)$ and the following hold*

- Completeness. *For all $(x, y) \in \mathcal{R}$, we have Pr$[(P, V)(x, y) = 1] = 1$.*

- Validity. *For every PPT ITM P', if Pr*$[(P', V)((x, \overline{x}), (y, \overline{y})) = 1] = p > \epsilon$*, then there exists a PPT resetting ITM E such that*

$$Pr[E^{P'}(x, \overline{x}) = (y, \overline{y}) \wedge (x, y) \in \mathcal{R} \wedge \bigwedge_{i=1}^{m}(x_i, y_i) \in \mathcal{R}_i] = p - \epsilon$$

- Zero-knowledgeness. *For any PPT ITM V^*, there exists a PPT ITM simulator M^* such that the following two ensembles are indistinguishable.*
  - $\{(P, V^*)(x)\}_{x \in L_{\mathcal{R}}}$
  - $\{M^*(x)\}_{x \in L_{\mathcal{R}}}$
- Infeasibility. *It is infeasible for any PPT ITM P to conduce the proof satisfying validity property above without the knowledge to $(y, y_1, y_2, \ldots, y_m)$.*

The infeasibility is a binding property in the sense that x is formed with the knowledge of witnesses for each aggregating element. When the languages L_i are all the same as L, then we say that L has an aggregated proof of knowledge.

The noninteractive case can be defined similarly as the interactive one. Where the knowledge extractor needs to reset the common reference strings to extract the knowledge.

Note that the aggregate ZKPK can be trivially implemented with repetition of proofs for each of aggregated elements. In that case the round complexity of aggregated proof would depend on the number of aggregated elements. However, in many applications, an aggregate proof that is sublinear or more efficient is expected. In the protocols following in the next section we show how we construct efficient aggregate ZKPK.

3.3 Aggregate Zero-Knowledge Proof Protocols

In this section we present our protocols to enable principals to enroll with registrars, and illustrate how service providers can verify the identity attributes using privacy preserving multi-factor identity verification mechanism. More specifically, we provide detailed

Table 3.1 Roadmap of the identity protocols with the identity assurance functions.

Phase	Functions	Cryptographic Protocols
Enrollment	LocalConsistency; FederationDuplicate Detection; ExternalValidation	Protocol 1 §3.3.1
Update	LocalConsistency; FederationDuplicate Detection	Protocol 1 §3.3.1
Usage	ExternalValidation	Protocols 2, 3(a,b) §3.3.2, Protocols 4(a,b) §3.3.3

protocols based on aggregate ZKPK that are employed in the enrollment of the SIT identifiers, and the signing of the commitments. We also show how such commitments can be used in the verification phase. For clarity in Table 3.1 we provide a roadmap indicating how the identity assurance functions as defined in Chapter 2 Section 2.3, and cryptographic protocols are used in the different stages. Our approach is based on aggregation techniques of committed values to provide flexible and efficient zero-knowledge proofs. We also extend the aggregation protocols to provide signature verification with hidden commitments in Section 3.3.3.

3.3.1 Commitments and Signatures at Enrollment

As stated in Chapter 2, at the time of enrollment, for each SIT strong identifier, the strong identifier binder (denoted by $\widehat{M}$) is needed as well as the semantically secure commitment (denoted by M). M is signed and stored in the IdR, while $\widehat{M}$ is used by registrars to detect duplicates. In the following protocol we show how M can be created by the principal interacting with the issuer. We also show how the principal can prove that the two refer to the same secret m. Finally, we illustrate how M is signed by the registrar. Our enrollment scheme provides one with the capability of verifying later on in an aggregate fashion

several strong identifiers issued by different issuers. Formally, the protocol is composed of the following steps.

Protocol 1: Computing a signature on an information-theoretic hiding committed value.

1. *Registrar's parameters.* The registrar runs generation algorithm GenKey on input 1^k to generate the public parameters: a prime q of length k, three groups G_1, G_2, G_T of order q. Two generators g_1, h_1 in G_1 are specified such that $\log_{g_1} h_1$ is unknown. An additional generator $g_2 \in G_2$ is needed, as well as a secret key $\chi \in \mathbb{Z}_q$ and the public key $v = g_2^{\chi}$. The resulting set of public parameters is $(G_1, G_2, G_T, g_1, h_1, g_2, v)$.

2. *Commitment of a value* $m \in \mathbb{Z}_q$. The principal chooses a value $r \in \mathbb{Z}_q$, and computes $M = g_1^m h_1^r$. The first time this commitment is computed by the user when the issuer is constructing the certificate $\Upsilon = Cert_{Issuer}\{M||\widehat{M}\}$ based on blind signatures. We refer the reader to [41, 42, 47, 48] for details on the blind signature based certificate issuance. We focus on the how the SIT strong identifiers in these certificates are enrolled and used in the federation.

   Without loss of generality, we consider the commitment M constructed by the user, when interacting with the issuer, to be the same one which is committed to the registrar[1].

3. *Zero-knowledge proof of committed value.* The principal gives a ZKPK of opening of the commitment M to the registrar.

$$PK\{(\alpha, \beta)\colon M = g_1^{\alpha} h_1^{\beta},\ \alpha, \beta \in \mathbb{Z}_q\}$$

[1]At the time of enrollment with the registrar the user could construct a new semantically secure commitment of the form $M' = g_1^m h_1^{r'}$. The two commitments can be trivially proven to be on the same strong identifier m using the following proof of knowledge:

$$PK\{(\alpha, \beta, \gamma)\colon M = g_1^{\alpha} h_1^{\beta} \wedge M' = g_1^{\alpha} h_1^{\gamma},\ \alpha, \beta, \gamma \in \mathbb{Z}_q\}$$

4. *Signing of a committed value*. After performing the security checks on the committed value (namely the local consistency and federation duplicate detection), the registrar executes the $Sign$ algorithm on the commitment M to output M^{χ} as the signature where χ is the secret key of the registrar.

3.3.2 Multi-factor Identity Verification

Assume that principal P requests a service from a SP which requires P to first authenticate by proving that it knows how to open a specified set of commitments. To indicate this set of commitments a set of tags is given that is denoted by ψ_{proof}. Moreover, to be authorized for the service the SP usually requires the principal to open or reveal in clear values some of the strong identifiers in its IdR. We denote this set of tags as ψ_{open}. The signatures and public parameters are retrieved from the principal's IdR, introduced in Chapter 2 Section 2.3.1. As multiple commitments have to be verified and proven, and can change according to the specific SPs identity verification policy, ψ_{proof} and ψ_{open} are not pre-determined. As such, we need an aggregation technique to combine any given combination of commitments and signatures for verification. For instance, in Chapter 2 Example 4, when Alice requests for service from SP-Shop, she needs to prove knowledge of $\psi_{proof} = \{SSN, CCN\}$ and provide in clear $\psi_{open} = \{CCN\}$. In what follows we illustrate how this can be achieved. Precisely, Protocols 2 and 3 provide aggregate proof of knowledge of the commitments corresponding to ψ_{proof} and ψ_{open} respectively. The protocols are two-party computations, in which the principal is the prover and the SP is the verifier (we use the two terms interchangeably).

Protocol 2: Proving aggregated signature on committed values. The principal performs the ZKP of the aggregated commitments corresponding to the tags given in ψ_{proof} and aggregated signature for verification.

1. *Principal's aggregation*. Let $\sigma_1, \sigma_2, \cdots, \sigma_t$, be the signatures corresponding to the tags in ψ_{proof}. The principal aggregates the signatures into $\sigma = \prod_{i=1}^{t} \sigma_i$, where σ_i is

the signature of committed value $M_i = g_1^{m_i} h_1^{r_i}$. It also computes $M = \prod_{i=1}^{t} M_i = g_1^{m_1+\cdots+m_t} h_1^{r_1+\cdots+r_t}$. Finally the principal sends σ, M, M_i, $1 \leq i \leq t$ to the verifier.

2. *Zero-knowledge proof of aggregate commitment*. The principal and the verifier SP carry out the following ZKP protocol:

$$PK\left\{(\alpha, \beta)\colon M = g_1^{\alpha} h_1^{\beta},\ \alpha, \beta \in \mathbb{Z}_q\right\}$$

3. *Verification of aggregate signature*. After the verifier accepts the zero-knowledge proof of the commitments, it checks if the following verifications succeed:

$$M = \prod_{i=1}^{t} M_i \qquad \text{and} \qquad e(\sigma, g_2) = e(M, v)$$

Only if steps 2 and 3 are successful, the SP will consider the signatures as valid. Step 2 provides an efficient approach to perform the ZKPK for each M_i in an aggregated manner that avoids carrying out a proof for each of the M_i's. Similarly, the aggregate signature in step 3 provides an efficient approach to check the signature for each of the commitment indicated in ψ_{proof}.

Protocol 3a: Opening the committed value. To satisfy SPs request to open in clear the principal's strong identifiers and verify the corresponding signatures, the principal has to show the corresponding values along with the commitments as given in ψ_{open}, as well as the aggregated signature.

The protocol relies on a random oracle hash function H which is known to all entities. Formally,

1. *Principal's aggregation and preparation*. Upon SPs requirement to open in clear $m_1, m_2, \cdots, m_t$

   (a) The principal aggregates the signatures $\sigma_1, \sigma_2, \cdots, \sigma_t$ into $\sigma = \prod_{i=1}^{t} \sigma_i$ where σ_i is the signature of committed value $M_i = g_1^{m_i} h_1^{r_i}$.

(b) Using H, principal computes the random values $(x_1, x_2, \ldots, x_t) = H(m_1 \parallel \cdots \parallel m_t \parallel M_1 \parallel \cdots \parallel M_t)$[2]. It also computes $r = \sum_{i=1}^{t} r_i x_i$ which is used in the zero-knowledge proof in the next step.

The principal sends $(m_1, \ldots, m_t), (x_1, \ldots, x_t), (\Upsilon_1, \ldots, \Upsilon_t)$[3] to the verifier SP.

2. *Zero-knowledge proof of aggregate commitment*. The principal and SP compute $M = \prod_{i=1}^{t} M_i^{x_i}$ and carry out the following ZKPK:

$$PK\left\{(\alpha, \beta)\colon M = g_1^{\alpha} h_1^{\beta},\ \alpha, \beta \in \mathbb{Z}_q\right\}$$

3. *Verification of aggregate signature*. After the verifier receives $M_1, M_2, \cdots, M_t$, and accepts the zero-knowledge proof of the commitments, it checks if

$$M = \prod_{i=1}^{t} M_i \qquad \text{and} \qquad e(\sigma, g_2) = e(M, v)$$

Only if all above checks are successful, SP validates the signatures and the strong identifier values $m_1, m_2, \cdots, m_t$.

Steps 1-3 are executed to ensure that the opened values $m_1, m_2, \cdots, m_t$ are the same as the ones originally committed $\{M_1, \ldots, M_t\}$. The knowledge of the m_i's is not sufficient to perform a successful proof of knowledge since also the committed random value r_i is needed to complete the proof. This requirement prevents possible misuse of the m_i's by the verifier SP. Note also that step 1b) corresponds to the challenge creation in a random oracle model, to enable a non-interactive ZKP according to the Fiat-Shamir [62, 63] paradigm.

[2]Here the random function H is from $\{0,1\}^*$ onto $\{0,1\}^{ck}$, where c is a constant, k is the security parameter. For any $x \in \{0,1\}^*$, let $y = H(x)$. For any given $ck > t > 0$, let $m = \lfloor |y|/t \rfloor$, to denote x_i is substring in y of length m for $1 \leq i \leq t-1$, and x_t is the suffix of y with length of $|y| - (t-1)m$, such that $y = x_1 x_2 \ldots x_t$. We denote it as $(x_1, \ldots, x_t) = H(x)$.

[3]Note that $\Upsilon_i = Cert_{Issuer}\{M_i || \widehat{M_i}\}$.

Moreover, based on the provided certificates $\Upsilon_i = Cert_{Issuer}\{M_i||\widehat{M_i}\}$ the association of M_i and $\widehat{M_i}$ can be determined for correctness.

Protocol 3b: Hidden Strong Identifier Validation. In the following protocol we consider the specific case of transactions where the actual values of strong identifiers are not required to be released to the SP. For instance in Chapter 2, Example 4, say the SP-Shop does not really need the actual value of CCN because the issuer of the CCN, possibly a bank, based on the required information can credit SP-Shop with the required amount of money. In the protocol the strong identifier is revealed only to the issuer of that identifier and the strong identifier binders are sent to the SP instead of the clear values. Moreover, an additional cryptographic token is passed to the SP that is in turn forwarded to the issuer. Here, we assume that the principal knows the public key of the issuer. Formally,

1. *Principal's aggregation*. Upon SPs requirement to provide strong identifier binders $m_1, m_2, \cdots, m_t$, the following steps are executed:

   (a) The principal aggregates the signatures $\sigma_1, \sigma_2, \cdots, \sigma_t$ into $\sigma = \prod_{i=1}^{t} \sigma_i$ where σ_i is the signature of committed value $M_i = g_1^{m_i} h_1^{r_i}$.

   (b) The principal retrieves $\widehat{M_i} \in \Upsilon_i = Cert_{Issuer}\{M_i||\widehat{M_i}\}$ for $1 \leq i \leq t$ and $r = \sum_{i=1}^{t} r_i x_i$ which is used in the ZKP in the step 2. Using H, it also computes the random values $(x_1, x_2, \ldots, x_t) = H(\widehat{M_1} \parallel \cdots \parallel \widehat{M_t} \parallel M_1 \parallel \cdots \parallel M_t)$.

   (c) The principal constructs the following message for each issuer.

   $$\mathsf{Enc}_{Issuer}(\{m_1, \ldots, m_t, timestamp\})$$

   The principal sends $(\Upsilon_1, \ldots, \Upsilon_t)$, σ and $\mathsf{Enc}_{Issuer}(\{m_1, \ldots, m_t, timestamp\})$ to SP.

2. *Zero-knowledge proof of aggregate commitment*. Based on the provided certificates $\Upsilon_i = Cert_{Issuer}\{M_i||\widehat{M_i}\}$ the association of M_i and $\widehat{M_i}$ is determined. For the proof

of knowledge of the strong identifiers and associated secrets, the principal and SP compute $M = \prod_{i=1}^{t} M_i^{x_i}$ and carry out the following ZKPK:

$$PK\left\{(\alpha, \beta)\colon M = g_1^{\alpha} h_1^{\beta},\ \alpha, \beta \in \mathbb{Z}_q\right\}$$

3. *Verification of aggregate signature*. The principal sends σ to the SP which can verify the signature as follows:

$$M = \prod_{i=1}^{t} M_i \qquad \text{and} \qquad e(\sigma, g_2) = e(M, v)$$

Only if steps 2 and 3 are valid, validator SP will accept the truth of the signatures and send the message.[4] SP will append all the strong identifier binder numbers to the message for verification by the issuer as follows:

$$\mathsf{Enc}_{Issuer}(\{m_1, \ldots, m_t, timestamp\}), (\widehat{M_1}, \cdots, \widehat{M_t})$$

The issuer can then verify each of the m_i's for its validity as well as the freshness of the message using the $timestamp$. It can also check that m_i, $1 \leq i \leq t$, corresponds to the binder numbers as checked by the SP.

3.3.3 Signature Verification with Hidden Commitments

The tags associated with committed strong identifiers may potentially leak information about the individual. For example, if a SSN number is enrolled it would imply that the individual has some source of income within the U.S. This may be not be acceptable in some scenarios, as highlighted by next example.

Example 8 Consider a registrar R_{govt} that enrolls only government officials and requires high identity assurance for each of the commitments it signs. The commitments of the in-

[4] If more than one registrars signature is involved, more bilinear mapping computations are involved in the step 3.

dividuals in R_{govt} may correspond to the role of the individual in the organization. Suppose principal P_A has enrolled its *"secret service officer ID number"* with R_{govt} and has received a signed commitment corresponding to it. Consider now a hotel Ht which provides discounts to government officials. P_A while booking a room at Ht wants to apply for the discount. For this purpose P_A needs to prove the commitment signed by R_{govt}. If the commitment and the corresponding tags are given in clear, they will leak information regarding P_A being a secret service officer. Therefore, it is desirable that P_A be able to prove that it has enrolled *some* identifier with R_{govt} without revealing the exact commitment or the tag associated with it.

The above example can be generalized to the case where multiple commitments should be proven issued by a known registrar without actually disclosing the values of the commitment itself or the corresponding tags. To achieve this feature we introduce a new cryptographic primitive.

Protocol 4a: Integrating the zero-knowledge proof into the verification.

1. *Principal's aggregation*. Upon SPs requirement to prove $\sigma_1, \sigma_2, \cdots, \sigma_t$, the principal aggregates the signatures into $\sigma = \prod_{i=1}^{t} \sigma_i$ where σ_i is the signature of committed value $M_i = g_1^{m_i} h_1^{r_i}$. The principal also computes $m = m_1 + \cdots + m_t \pmod q$, $r = r_1 + \cdots + r_t \pmod q$.

2. *Zero-knowledge proof of aggregate commitment*. The principal sends σ to SP, and carries out the following ZKP protocol with SP:

$$PK\left\{(\alpha, \beta) \colon e(\sigma, g_2) = e(g_1, v)^{\alpha} e(h_1, v)^{\beta},\ 0 < \alpha, \beta < q\right\}$$

Note that the only information sent by the principal is σ, while in Protocol 3 also the tags and the commitments were sent. If the above checks are valid, verifier SP will validate the signatures.

Protocol 4b: Zero-knowledge proof the aggregated signature.

Protocol 4a is a secure protocol that hides the tags and commitments. However, it is not a ZKP protocol because different instances of the signature verification performed by the same principal can be linked by the SP because the signatures themselves are deterministic. Moreover, if the principal had revealed in an earlier transaction, its M_i (and possibly the tag associated with it) then the SP can link the σ_i with it. To address this shortcoming we provide a protocol variant in which even the actual signature is not revealed. More specifically, a randomized signature is used to verify that the original signature has been issued by a given registrar. As the signatures are randomized and the proof of validity is zero-knowledge, one signature cannot be distinguished from the other. The succinct ZKPK is to convince the verifier of the possession of knowledge of one signature on a committed value, rather than which one it is. The final submitted value is independent of any of the actual signatures. Therefore it is necessary that only one signature be verified. Any further verification of additional randomized signatures does not provide any additional information. This protocol has the advantage of assuring that a principal remains unlinkable and anonymous even if it had initially revealed its strong identifiers and commitments to the verifying SP.

1. *Principal's hiding*. Upon SPs requirement to prove a signature σ, principal chooses $r \in \mathbb{Z}_q$ at random, and sends the messages $\delta := \sigma^r$ to SP.

2. *Zero-knowledge proof of aggregate commitment*. The principal carries out the following zero-knowledge proof protocol with the verifier SP:

$$PK\left\{(\alpha, \beta)\colon e(\delta, g_2) = e(g_1, v)^{\alpha} e(h_1, v)^{\beta},\ 0 < \alpha, \beta < q\right\}$$

3.4 Analysis

In this section we analyze our solution. We first provide a formal analysis of the security of the cryptographic protocols introduced in Section 3.3. We then evaluate the computational complexity of the main protocols characterizing our approach. Finally, based on

the properties of our protocols and on the identity assurance methodologies presented in Chapter 2 Section 2.3, we briefly analyze how identity theft prevention is achieved in the resulting identity system.

3.4.1 Security Analysis of the Protocols

Before proving the security properties of our protocols, we identify the properties that characterize the cryptographic techniques used. The security of such cryptographic techniques relies on the assumption of the co-gap Diffie-Hellman (co-GDH) problem [22], which is summarized as follows.

For multiplicative cyclic groups G_1, G_2, G_T of order q, let g_1 be a generator of G_1 and g_2 be a generator of G_2. Let ψ be a computable isomorphism from G_1 to G_2, with $\psi(g_1) = g_2$ and e a computable bilinear map e: $G_1 \times G_2 \rightarrow G_T$. ψ and e can be computed efficiently. The co-GDH gap problem is relating two problems used in cryptography which are as follows:

Decisional Co-Diffie-Hellman problem: Given $\langle g_1, g_2, g_1^a, g_2^b, g_2^c \rangle$ for some $a, b, c \in \mathbb{Z}_q^*$, to decide if $c = ab \mod q$.

Computational Co-Diffie-Hellman problem: Given $\langle g_1, g_2, g_1^a, g_2^b \rangle$ for some $a, b \in \mathbb{Z}_q^*$, to compute $g_2^{ab} \in G_2$.

Groups G_1, G_2 are said to be **Co-GDH groups** if there exists an efficient algorithm to solve the Decisional Co-DH problem and there is no polynomial-time (in $|q|$) algorithm to solve the Computational Co-DH problem. The existence of a cryptographic bilinear map ensures the existence of Co-GDH groups.

As the discrete logarithm assumption is implied by the co-GDH assumption, the results stated in the next lemma concerning the ZKPs appearing in Protocols 1, 2, 3a, 3b and 4a are derived from [61, 64].

Lemma 3.4.1 *Let G_1, G_2 be Co-GDH groups of prime order q with respect to generators $g_1 \in G_1$ and $g_2 \in G_2$. Let $h_1 \in G_1$ be a generator with $\log_{g_1} h_1$ unknown. The ZKPK appearing in Protocols 1, 2, 3a, 3b and 4 hold true for the specified parameters. More precisely:*

1. *Step 3, Protocol 1, and step 2 in Protocol 3b are ZKPs of knowledge of the values* $m_i, r_i \in \mathbb{Z}_q$ *such that the same* m_i *is committed in* M_i *and its strong identifier binder is* $\widehat{M_i}$.

2. *Step 2 in Protocol 2 is a ZKPK of the values* $\sum m_i \mod q$, $\sum r_i \mod q$.

3. *Step 2 in Protocols 3a and 3b is a ZKPK of the values* $\sum(m_i \times x_i) \mod q$, $\sum(r_i \times x_i) \mod q$ *where* $x_i \in \mathbb{Z}_q$ *are random challenges.*

4. *Step 2 in Protocol 4a is a ZKPK of the values* $\sum m_i \mod q$, $\sum r_i \mod q$ *satisfying the signature verification relation.*

We now show that all protocols are two-party secure computations. Security is ensured by proving *correctness* and *unforgeability* of each protocol.
Correctness of protocols means that honest users can, with correct data, carry out the protocols successfully, while unforgeability guarantees that an adversary, with forged data, cannot execute the protocols successfully. Our results on unforgeability for Protocol 2 are derived from Lemma 3.4.3.
Proving the security of the first protocol is straightforward. The following lemma is given.

Lemma 3.4.2 *In Protocol 1, let* G_1, G_2*, be Co-GDH groups of prime order* q *with respect to generators* $g_1 \in G_1$ *and* $g_2 \in G_2$*. Let* $h_1 \in G_1$ *be a generator, with* $\log_{g_1} h_1$ *unknown. The protocol is secure.*

The truth of Lemma 3.4.2 is based on the statistical hiding and computational binding properties of Pedersen commitments. Therefore, signatures and aggregation computed on such commitments will continue to hold those properties. The independent techniques employed in this protocol are conventional, and have been investigated separately in several papers [46, 65–67]. The correctness proofs are similar to the ones elaborated in Theorem 3.4.4 and are therefore omitted.

Lemma 3.4.3 (Unforgeability of Aggregation of Pedersen Commitment) *Let* G *be a group of prime order* q*, in which the discrete logarithm is hard to compute. Elements*

$g, h \in G$ are generators with $\log_g h$ unknown. $M_i = g^{m_i} h^{r_i}$ are Pedersen commitments to messages $m_i \in \mathbb{Z}_q$, and random numbers $r_i \in \mathbb{Z}_q$, with $1 \leq i \leq t$. Let $M = \prod_{i=1}^{t} M_i$. Then, it is infeasible, given only $M_1, M_2, \ldots, M_t$, to compute m, r such that $M = g^m h^r$ if at least one of m_i or r_i is unknown.

Proof Suppose that $m_1, \ldots, m_{t-1}$ and $r_1, \ldots, r_t$ are known, and m_t is unknown. If adversary can compute $m, r \in \mathbb{Z}_q$, where $m = \sum_{i=1}^{t} m_i, r = \sum_{i=1}^{t} r_i$ such that $M = g^m h^r$, then it can get $g^{m_t} g^{m_1+m_2+\cdots+m_{t-1}} h^{r_1+\cdots+r_t} = g^m h^r$. This means that $g^{m_t} = g^{m-m_1-\cdots-m_{t-1}} h^{r-r_1-\cdots-r_t}$, which implies $m_t \equiv m - m_1 - \cdots - m_{t-1} \mod q$ and $r \equiv r_1+\cdots+r_t \mod q$. This in turn implies that the adversary can solve the discrete logarithm $g^{m_t} = M/(g^{m_1+\cdots+m_{t-1}} h^r)$ with respect to g. As m_t is an arbitrary element in $\mathbb{Z}_q$, that is contradictory with respect to the discrete log problem (DLP) assumption. ■

Theorem 3.4.4 *For co-GDH groups G_1, G_2, Protocol 2 is a secure two-party computation.*

Proof We show that the prover needs to know all the committed values and that the associated signatures need to be valid to successfully execute the protocol.

Correctness: Let $M_i = g_1^{m_i} h_1^{r_i}$, $\sigma_i = M_i^{\chi}$, then $M = \prod_{i=1}^{t} M_i = \prod_{i=1}^{t} g_1^{m_i} h^{r_i} = g_1^m h_1^r$, where $m = \sum_{i=1}^{t} m_i$, $r = \sum_{i=1}^{t} r_i$. The prover is able to execute

$$PK\left\{(\alpha, \beta) \colon M = g_1^{\alpha} h_1^{\beta},\ \alpha, \beta \in \mathbb{Z}_q\right\}$$

with the knowledge of $\alpha = m$ and $\beta = r$ according to Lemma 3.4.1.

To prove correctness for step 3 of the protocol, which verifies the validity of the aggregated signature, we note that $\sigma = \prod_{i=1}^{t} \sigma_i = \prod_{i=1}^{t} M_i^{\chi}$, and

$$e(\sigma, g_2) = e\left(\prod_{i=1}^{t} M_i^{\chi}, g_2\right) = \prod_{i=1}^{t} e(M_i, g_2)^{\chi} = \prod_{i=1}^{t} e(M_i, v) = e(M, v).$$

Unforgeability: We prove this property by showing that if the prover does not know even one of the messages $\{m_i\}_{1 \leq i \leq t}$ and $\{r_i\}_{1 \leq i \leq t}$, OR one of σ_i, $1 \leq i \leq t$, is not valid, then the protocol fails.

If the prover does not know all the secrets and the proof is executed successfully, this would mean that there exists a knowledge extractor that can extract two values m' and r' such that $M = g_1^{m'} h_1^{r'}$. However, according to Lemma 3.4.3 this is infeasible.
For the case in which any one of the signatures is not valid, the step 3 of the protocol will not succeed because of the security of the aggregated signature as given in [22]. ■

Theorem 3.4.5 *For co-GDH groups G_1, G_2, the following results hold:*

1. *Protocol 3a is a secure two-party computation. It guarantees that 1) principal has knowledge of values r_i, 2) the values m_i are correctly committed in M_i, and 3) signatures $\sigma_1, \sigma_2, \ldots, \sigma_t$ are valid.*

2. *Protocol 3b is a secure two-party computation. It guarantees that 1) principal has knowledge of values r_i, 2) the values committed in M_i and $\widehat{M_i}$ are the same, and 3) signatures $\sigma_1, \sigma_2, \ldots, \sigma_t$ are valid.*

Proof We show that it is correct and unforgeable.

Correctness: We prove that for the honest prover, with values $m_1, \cdots, m_t, M_i = g^{m_i} h^{r_i}$, the protocols execute correctly. After computing the values $(x_1, \cdots, x_t)$ the prover calculates $M = \prod_{i=1}^{t} M_i^{x_i} = g_1^m h_1^r$. Here, $m = \sum_{i=1}^{t} m_i x_i$ and $r = \sum_{i=1}^{t} r_i x_i$. From this the prover is able to carry out the ZKPK $PK\{(\alpha, \beta) : M = g_1^{\alpha} h_1^{\beta}\}$. As such, only by knowing all $m_i, 1 \leq i \leq t$ and $r_i, 1 \leq i \leq t$ the correct value m and r can be computed and substituted for α and β. Items 1) and 2) of the thesis are thus proved for both Protocols 3a and 3b. The association of M_i and $\widehat{M_i}$ is clearly determined based on the provided certificates $\Upsilon_i = Cert_{Issuer}\{M_i || \widehat{M_i}\}$.

The correctness of the signatures $\sigma_1, \sigma_2, \ldots, \sigma_t$ is derived from the validity of aggregated signature σ. We omit the correctness proof for σ because it is similar to the proof given for Theorem 3.4.4.

Unforgeability:

1. *Protocol 3a* We show by contradiction that the successful execution of the protocol guarantees that a prover cannot forge even one of r_i or m_i.

Assume that an adversary executes successfully the protocol based on its knowledge of $r_2, \cdots, r_t$ and $m_1, \cdots, m_t$ and that it does not know r_1. To execute step 2 of Protocol 3a (3b), a zero-knowledge extractor that extracts m and r has to exist. Thus the adversary can feasibly compute r and, from $r = \sum_{i=1}^{t} x_i r_i \mod q$, it can deduce r_1. Thus it can feasibly compute r_1 which is a contradiction with respect to the assumption that r_1 is unknown. The same argument holds true for m.

Unforgeability of m_i means that it is infeasible for a prover to reveal a set of $\{m'_1, \cdots, m'_t\}$ which is not exactly the same as $\{m_1, \cdots, m_t\}$ corresponding to the original identifiers committed in $M_1, \cdots, M_t$ and successfully execute the protocol. In this case at least one m'_i with $m'_i \neq m_i \mod q$ exists which would result in the random challenges to be calculated as $(x_1, \cdots, x_n) = H(m'_1 \parallel \cdots \parallel m'_t \parallel M_i \parallel \cdots \parallel M_t)$. Step 2 of Protocol 3a performs the ZKPK, showing that $M = g_1^m h_1^r, r = \sum_{i=1}^{t} x_i r_i \mod q$, $M = \prod_{i=1}^{t} M_i^{x_i} = g_1^m h_1^r$. Here, $m = \sum_{i=1}^{t} x_i m_i$, $m' = \sum_{i=1}^{t} x_i m'_i$. Since $\log_{g_1} h_1$ is unknown, $g^m = g^{m'}$ implies $m - m' = 0 \mod q$. That is, $\sum_{i=1}^{t} x_i(m'_i - m_i) = 0 \mod q$. Because there exists at least one i such that $m'_i \neq m_i \mod q$, and since $(x_1, \cdots, x_n)$ is random, it is infeasible that $\sum_{i=1}^{t} x_i(m'_i - m_i) = 0 \mod q$.

2. *Protocol 3b* The same reasoning as before applies to Protocol 3b, with the only difference that SP does not explicitly know the values $m_1, m_2, \ldots, m_t$.
   By using messages $\mathsf{Enc}_{Issuer}(\{m_1, \ldots, m_t, timestamp\}), (\widehat{M_1}, \cdots, \widehat{M_t})$, the issuer will check if the $\widehat{M_i}$ binder corresponds to the strong identifier provided. If they are all valid, and from the correctness of the proof, we know that m_i is the value committed in M_i, $1 \leq i \leq t$, which were signed by registrar.

■

Theorem 3.4.6 *For co-GDH groups G_1, G_2, Protocol 4a is a secure two-party computation in random oracle model.*

Proof We show that it is correct and unforgeable.

Correctness: From the signatures $\sigma_1, \sigma_2, \ldots, \sigma_t$ assigned by the registrar for messages $M_1 = g_1^{m_1} h_1^{r_1}, \ldots, M_t = g_1^{m_t} h_1^{r_t}$, the principal computes

$$\sigma = \prod_{i=1}^{t} \sigma_i = \prod_{i=1}^{t} M_i^{\chi} = \prod_{i=1}^{t} g_1^{m_i \chi} h_1^{r_i \chi} = g_1^{m\chi} h_1^{r\chi}$$

Where $m = \sum_{i=1}^{t} m_i,\ r = \sum_{i=1}^{t} r_i$. Then

$$e(\sigma, g_2) = e(g_1^{m\chi} h_1^{r\chi}, g_2) = e(g_1^m h_1^r, g_2)^{\chi} = e(g_1, v)^m e(h_1, v)^r$$

where χ and $v = g_2^{\chi}$ are respectively the private and public keys of the registrar. The principal is able to successfully carry out the ZKPK m, r as given in step 2 of the protocol.

Unforgeability: The successful protocol execution should guarantee that the prover has valid signatures σ_i and knowledge of all m_i committed in M_i. If a knowledge extractor exists for the ZKPK at step 2 that extracts two message m' and r', such that $e(\sigma, g_2) = e(g_1, v)^{m'} e(h_1, v)^{r'}$, then it would mean that $e(\sigma, g_2) = e(g_1^{m'\chi} h_1^{r'\chi}, g_2)$. More specifically, $e(g_1^{m\chi} h_1^{r\chi}, g_2) = e(g_1^{m'\chi} h_1^{r'\chi}, g_2)$. Since G_1, G_2 are Co-GDH groups, it implies that the principal knows the values m and r. By Lemma 3.4.3, we know that the prover has the knowledge of all the values m_i committed in the messages M_i $1 \leq i \leq t$. Thus, the validity of signatures $\sigma_1, \ldots, \sigma_t$ is obtained from the security of aggregation signature [22]. ■

Theorem 3.4.7 *Protocol 4b is a ZKP of a signature on a message under signature scheme of Protocol 1.*

Proof To show the zero-knowledge property, we construct a simulator S as follows. Because the message that the principal sent in the first step is independent of any actual signature, S randomly chooses $s_1, s_2 \in \mathbb{Z}_q$, and forms $g_1^{s_1} h_1^{s_2}$ that has the following property:

$$e(g_1^{s_1} h_1^{s_2}, g_2) = e(g_1, g_2)^{s_1} e(h_1, g_2)^{s_2} = e(g_1, v)^{s_1/\chi} e(h_1, v)^{s_2/v}$$

Table 3.2 Comparison on the number of exponentiations for proving t factors.

		Protocol 2	Protocol 3a	Protocol 3b	Protocol 4a	Protocol 4b
Our Protocols	provers	$2+2$	$3+2$	$3+2$	2	2
	verifiers	3	$2t+3$	$2t+3$	3	3
Without Aggregation	provers	$2t$	$4t$	$4t$	$2t$	$\times$
	verifiers	$3t$	$5t$	$5t$	$3t$	$\times$

The above results in the correct form of the required signature. Because in step 2, the principal and SP execute a ZKP, it follows that there exists a simulator S' for that step. When S' is run, it is easy to deduce that the simulator S constructed is the zero-knowledge simulator for the protocol.

Next, we show that Protocol 4b is a proof of knowledge. Suppose a prover can give an acceptance proof following the protocol, the knowledge extractor for it will obtain values $m_0, r_0 \in \mathbb{Z}_q$, such that

$$e(\sigma, g_2) = e(g_1, v)^{m_0} e(h_1, v)^{r_0} = e(g_1, g_2)^{m_0 \chi} e(h_1, g_2)^{r_0 \chi} = e((g_1^{m_0} h_1^{r_0})^{\chi}, g_2)$$

This forms a signature pair $(g_1^{m_0} h_1^{r_0}, \sigma)$. ■

3.4.2 Complexity Evaluation of the Protocols

Our ZKPK is based on the difficulty of the discrete logarithm, which is implied by the assumption of co-GDH groups. To compute the proof of $PK\{(\alpha, \beta) : y = g^{\alpha} h^{\beta}, \alpha, \beta \in \mathbb{Z}_q\}$, five exponentiations are used [68]. If a separate proof of the knowledge for t commitments were used, then $5t$ exponentials would need to be computed. In some of our protocols, we reduce the number of exponentiations to a constant that does not depend on the number of commitments to be proved. In our protocols, principals always need

compute a constant number of exponentiations, while the verifier's computation of exponentials is mostly dramatically reduced (see Table 3.2). These simple considerations prove the efficiency and practical features of our approach. Table 3.2 reports a comparison of the exponentiations computed by the principals or provers and verifiers in our aggregate protocols and in the case when they are not aggregated.

As we adopted the Pedersen commitment and the short signature from [69], our signatures on commitments are short and the storage complexity is smaller than the ones computed with existing techniques [21]. As an example, even the simplest version of signature has a length three times than ours.

Camenisch *et al.* also considered signatures on the commitments on a set of messages (see [21], page 10 and Theorem 3.) Compared to their methods, our approach is more flexible in that whenever n messages are committed for a user, the user is able to to prove $2^n - 1$ many combinations of them, which does not appear possible in the scheme by Camenisch *et al.* Because we make use of the aggregation signatures developed in [22] to sign the Pedersen's commitments, the verification of the signature is more efficient than if the verification were executed separately.

Moreover, in our case, because the signatures stored in a particular IdR are assigned only by the registrar that enrolls it, the verification becomes even more cost effective. We compare to the case in which aggregated proofs are not used. Non-aggregated proofs need $2t$ many bilinear mapping computations for the verification of t signatures, while each of our protocols needs only 2 bilinear mappings, which is again a constant and is independent of the number of signatures proved.

To summarize: no matter how many factors need to be proved, with users having to perform only constant exponentials (at most five) in the proof, it is practical to execute our protocol in lightweight devices such as smart card, mobile devices and so on. No matter how many factors need to be proven, there are only two bilinear mapping computations needed in each protocol, and the number of exponentials for the verifier are dramatically reduced.

3.4.3 Security analysis of the Federation System

We discuss how our identity assurance techniques and cryptographic protocols together guarantee the security of the federated identity management system with respect to robustness and confidentiality. In our context, robustness means that no theft of identity attributes can be perpetrated within the federation. Confidentiality means that no unauthorized third party can gain access to the data exchanged during the registration and the usage protocol.

Robustness against theft of identity attributes. An important property that our protocol must ensure is that no matter how the federation entities might collude, it must not be possible for any entity to succeed in compromising use of identifiers of other principals. Thus, it must not be possible that a principal P uses a strong identifier m belonging to another principal $P' \in \mathcal{P}$ unless P can prove ownership of m as well. Further, the SP can ensure that an adversary will not succeed in using strong identifiers belonging to any other individual, even if such individual has never registered the identifier with the federation. To show robustness we focus on the most interesting misbehavior by the different entities.

(i) Dishonest principal P. At the time of registration, two possibilities arise: the first is the case in which P impersonates an already registered individual, $P' \in \mathcal{P}$, by trying to register m which is owned by P'. P fails registering m because the strong identifier binder $\widehat{M}$ is in fact already recorded by the FederationDuplicateDetection function (see Table 2.2), when P' enrolled it. The other possible case is that P is impersonating an individual not known to the federation by registering a strong identifier m. Here, theft by P is detected because ExternalValidation is executed for a minimum number of strong identifiers as defined by the federation. Protocols 2, 3 and 4 provide efficient and flexible approaches to perform the multi-factor identity verification at the time of usage. Each of them are secure as proved by Theorems 3.4.4 $-$ 3.4.7. Therefore impersonation can only be achieved with the compromise of all the required identifiers.

(ii) Honest principal P with dishonest registrar. Within the federation even a registrar cannot misuse the data, because it cannot prove the ownership of a valid IdR. This is ensured by the ZKPK protocol presented in Section 3.3. Another possible misbehavior of the

registrar not strictly related with identity theft is related with corrupting the IdR. Precisely, the value of one or more stored commitments in IdR may be changed to an incorrect value. However, because the principal generates these values independently from the registrar's input, such errors can be detected.

(iii) Honest principal P with dishonest SP. Even in case a dishonest SP attempts an identity theft, it cannot reuse the proofs or the signatures to prove ownership of the corresponding strong identifier. This condition holds even if the SP knows the actual value of the strong identifiers, because of the semantically secure Pedersen's commitment. Moreover as illustrated in Protocol 3b the $timestamp$ prevents any replay attack of a final token sent to the issuer.

Confidentiality. Confidentiality of strong identifiers is achieved through combination of PKI techniques and the security of the protocols. Precisely, confidentiality is achieved as follows. Concerning identifiers registration, as illustrated in Protocol 2, only the commitments of the strong identifiers are revealed. From Lemmas 3.4.1 and 3.4.2, it follows that the values of the strong identifiers in the commitments remain confidential. With respect to usage of identifiers our protocols preserve minimality, in that if the values of the strong identifiers are not required to be revealed at the time of usage, then as illustrated in Protocols 3b, 4a and 4b, we derive that the confidentiality of the strong identifier is assured. Concerning the confidentiality of weak identifiers and strong identifiers' tags, Protocol 4(a,b) provides an elegant way to hide the entire IdR. Moreover, subsequent usage of the the signatures cannot be linked in Protocol 4b that is proved in Theorem 5.6 part 2. Protocol 4(a,b) directly implies that SPs do not have access to the tags of the committed values and they cannot infer which strong identifiers have been committed.

3.5 Summary

The AgZKPK protocols presented in this chapter provide a flexible and privacy preserving methodology to perform multi-factor identity verification. We have shown that our aggregated multi-factors ZKPKs are more efficient than separate cases ZKPKs. More-

over users only need to calculate a constant number of exponentials, no matter how many signatures and commitments are to be proved. Our proof of knowledge of signature on commitment is more computationally and storage efficient than existing approaches [21]. Our aggregate proof is more flexible and requires a small amount of storage. The verification of the aggregated signature is also efficient. The small security parameters such as q used by the Weil or Tate Pairing [70] can be efficiently implemented better than RSA on small devices such as smart cards. Together with the additional composite protocols for maintaining identity assurance as introduced in Chapter 2, this forms an important part of the solution to prevent identity theft.

There are specific assumptions to consider while employing the AgZKPK protocols. First, the protocols are based on the discrete logarithm assumption. Second, it is assumed that a standards based IdM message exchange format [71] is employed to execute the protocols.

One key assumption for the correctness of AgZKPK protocols is that all the secrets associated with the principal's SIT identifiers that are required to create the proof are not compromised. Therefore it is required that the principals employ mechanisms to manage and secure the secrets. In addition, even as minimal information is revealed during the multi-factor verification, it may be possible for the verifier to infer additional information about a principal based on the principal's activities including authorization and service access.

A SP is assumed to define identity verification policies that require proofs of identity and ownership in addition to traditional forms of identity attributes. This was illustrated with representative examples with the AgZKPK protocol descriptions. Moreover the SP is assumed to have an AgZKPK verification component to perform the verification steps for the ZKPK and aggregate signatures as defined in the protocols.

4. BIOMETRIC IDENTITY VERIFICATION USING BIOMETRIC COMMITMENTS

In this chapter we extend the multi-factor identity verification requirement highlighted in Chapter 2 to include biometric identifiers as SIT identifiers. In general, biometric identifiers are verified using biometric verification systems that are automated methods for recognizing an individual based on some physiological and behavioral characteristics, such as fingerprints, voice, or facial features. Biometric verification[1] provides some inherent advantages as compared to other non-biometric identifiers because biometric characteristics correspond to a direct evidence of the personal identity versus possession of secrets that can be potentially stolen. Moreover, most of the times biometric enrollment is executed in-person and in controlled environments making it reliable for subsequent use [4].

Biometric verification poses several non-trivial security challenges because of the inherent features of the biometric data itself. Addressing these challenges is crucial for the large scale adoption of biometric verification, its integration with other verification techniques, and with access control systems.

In typical biometric verification systems, at the time of enrollment the individual's biometric sample is processed into a template to be used for subsequent verification attempts using biometric matching. This template is in the form of digital data and often stored in a database or on a token. Biometric matching is probabilistic in nature, which means that two samples of the same individual are never exactly the same. If the two samples are encrypted for security reasons, they need to be decrypted before they can be matched. This raises the issue of cryptographic key management to enable decryption of a stored biomet-

[1]Within the biometric community biometric authentication is more specifically referred to as either verification or identification. In verification user makes a claim to identity and then matching of the user sample presented to the system is executed on a one-to-one basis. Identification does not require a claim to identity; therefore the current sample is compared against a large number of templates in the database until a match is found. In this chapter we focus on biometric verification.

ric template, and also represents a point of vulnerability where if the cryptographic key is compromised then the confidentiality of the biometric template is also compromised. Unlike some password systems that perform a one-way hash function on the user input, biometric systems cannot rely on the same process. The reason is that the cryptographic hash values will never be the same for the reference template value stored at enrollment, and sample presented at verification. Additionally templates are often vendor-specific and therefore the interoperable use of such templates in a distributed system is difficult.

Biometric verification from an unsupervised location also presents the possibility for sensor spoofing attacks. The credibility of the output from a biometric matching process depends entirely on the integrity of the sample provided, and whether it is a true sample provided by the owner of the biometric characteristic. Older generation biometric capture devices were vulnerable to spoofing attacks, and there is extensive work on-going to mitigate biometric sensor spoofing [72, 73].

Biometric verification can be implemented through systems performing template matching either on the *server* or on the *client side*. Depending on where the matching of the biometric template is executed - at the server or at the client - different security problems arise. In the former case the main issues are related with the large scale and distributed management of biometric templates. The creation of a database of a particular biometric at the server should itself be secure and possibly decentralized. Also, such database may depend on a particular template creation and matching algorithm as well as hardware and thus may not be interoperable.

Additionally, storing biometric information in repositories along with other personally identifiable information raises several security and privacy risks [10]. These databases are vulnerable to attacks by insiders or external adversaries and may be searched or used outside of their intended purposes. It is important to note that if the stored biometric templates of an individual are compromised, there will be severe consequences for the individual because of the lack of revocation mechanisms for biometric templates.

Because of the security and privacy problems of server side matching, several efforts in biometric verification technology have tried to develop techniques based on client side

matching [74, 75]. Such an approach is convenient as it is relatively simple and cheap to build biometric verification systems supporting biometric storage at the client end able to support local matching. Nevertheless, systems of this type are not secure if the client device is compromised; therefore additional security mechanisms are needed.

Client side verification systems led to research on key generation mechanisms that use biometrics [74–80]. Key generation is generally executed by first extracting the biometric features from the biometric data based on a feature extraction module of the biometric verification system. Then, the biometric features are sent to the system specific key-generation module to generate a key, that we refer to as *biometric-key* (BK for brevity). The BK is never stored at any location and the key generation mechanisms should not allow re-generation of the BK without the individuals' real biometric. Note also, that the biometric template is not stored, therefore verification does not involve biometric matching and instead uses the BK. One main challenge in such an approach is to devise algorithms for reliable BK re-generation that is used for verification. Reliability is based on two specific properties, namely uniqueness and repeatability. Uniqueness of BK is required to ensure that two different individuals do not generate the same BK. Repeatability refers to the ability by an individual to re-generate its own BK.

Our scheme for key generation is developed based on singular vector decomposition (SVD) based image hashing techniques followed by support vector machine (SVM) based classification. Although several approaches have been taken to generate biometric keys, they are based on specific biometric features [80], and cannot be used for other types of biometrics. In our approach we generate keys of comparable bit lengths, using the generic image features of 2D images of biometric. More specifically, we show that our generic biometric key generation techniques work effectively for fingerprint images, iris images and face images. Through empirical analysis, we find substantial improvement in the performance with respect to false rejection rate and false acceptance rate as compared to the closely related schemes [81, 82]. Our generic image based approach is suitable for multimodal biometric systems [83]. Multimodal biometric systems utilize more than one physiological or behav-

ioral characteristic for enrollment and verification. Such capability is also inline with the multi-factor identity verification elaborated in Chapters 2.

A main advantage of our protocols is the privacy and security properties of the resultant biometric verification system. We show that the privacy of the biometric is preserved as the final BK does not reveal any information of the original biometric image. The key generation process includes pseudorandom values in several steps of the protocol. The pseudorandomness and specific properties of the SVD ensure that it is computationally hard for an attacker to retrieve the original image even if the BK is compromised. We also analyze several attack scenarios, including the case when all secrets on the device are compromised; even in this case it is computationally hard for the attacker to retrieve the BK.

Another main advantage is that we encode the BK into a cryptographic biometric commitment that is used in ZKPK at the time of verification, using protocols detailed in Chapter 3. This way the biometric identifiers can be used together with other SIT identifiers in multi-factor identity verification. Using such techniques the same BK can be used multiple times with the same or different verifying parties without the verifying party being able to link the transactions based on the cryptographic ZKPK proof of BK. It follows from the zero knowledge proof protocols that the cryptographic proofs cannot be replayed. As such the verifying party obtains no information about the characteristics of the real biometric based on the cryptographic proof. Moreover, using BK to construct biometric commitments eases the revocation and re-enrollment mechanisms of the BK.

The chapter is organized as follows. In Section 4.1 we provide an overview of our approach. In Section 4.2 we provide the biometric key generation protocol. Specifically in Section 4.2.2 we provide our hashing algorithm and in Section 4.2.3 we provide the SVM techniques we employ. This is followed by the experimental results of these Algorithms in Section 4.2.4. In Section 4.3 we provide a detailed analysis of our algorithms and approach followed by a summary in Section 4.4.

4.1 Overview of Approach

We begin with providing an overview of the biometric key (BK) life cycle, to clarify the main steps of a BK, from its generation to its dissolution. We describe the step by step process to be conducted for deploying a biometric key in a given organization having a finite set of individuals.

The first stage is the *configuration* stage, during which the BK generation algorithm is configured and tested. Configuration is based on BK-FEATURE-VECTOR which defines the features of the biometric that will be used for capturing the biometric characteristics. The BK-FEATURE-VECTOR and the BK generation algorithm are tested with a database of the potential individuals' biometric, to ensure that the *uniqueness* and *repeatability* properties of the resultant BK will be satisfied. Based on the test results, the algorithm parameters can be fixed, to ensure the highest possible accuracy and robustness of the biometric. An example to reflect such case is as follows.

Example 9 Consider a case where biometric key generation system is configured to generate keys from fingerprints using optical scanners. However, the individual using it instead has a thermal fingerprint scanner that outputs slightly different fingerprint images. In this case, the biometric system can be re-configured to set parameters suited better for the thermal scanners.

The parameters are determined at the configuration stage along with a basic classification model for the BK generation and is provided to the individuals to be used by their client devices.

The BK generation occurs at the time of the individual's enrollment. The enrollment consists of two phases as illustrated in Figure 4.1 both executed at the client device. Depending on the policy of the verifying party, enrollment can be executed either in-person at a physical location of the verifying party or online. If the verifying party wants to control which biometric is used then the enrollment must be in-person. However, if for the verifying party it is not relevant how the key is generated, the BK enrollment can be online. At

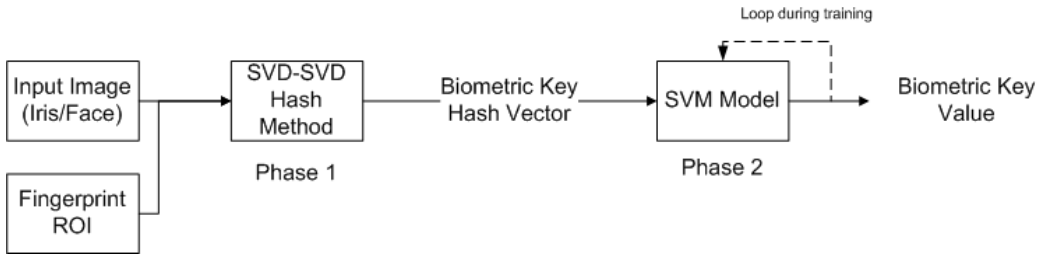


Fig. 4.1. Two main phases of the biometric key generation.

either enrollment, the individual records several readings of its biometric. The resultant biometric templates[2] are then used to provide a set of possibly different BK-HASH-VECTORs. A BK-HASH-VECTOR is a bit string that represents the biometric and is obtained from the biometric through an image hashing algorithm based on Singular Value Decomposition (SVD), as depicted in Phase 1 of Figure 4.1. In Phase 2, the resultant BK-HASH-VECTORs are classified to obtain a combination of classes (denoted as BK-CLASS-COMB) which represent the user's unique and repeatable BK. The classification, followed by choosing the combination of classes is based on Support Vector Machines (SVM). The generation process also returns some meta-data, referred to as BK-META-DATA that is used when re-generating the BK-CLASS-COMB. The BK-META-DATA consists of the classifier model and the pseudorandom secrets involved in the hashing algorithm.

The BK-CLASS-COMB is essentially the final BK that is used for identity verification. Specifically, identity verification is based on the aggregate ZKPKs which are presented in Chapter 3. In the context of the biometric identifiers, the BK is the private secret of the individual and is used with an additional random secret r to create an information theoretically secure Pedersens commitment [84]. This commitment is used to construct a ZKPK proof. This proof is sufficient for the purposes of verification as it corresponds to the biometric enrolled in the system. We refer to the commitment as BK-COMMIT, which is

[2]The digital representation of a biometric is called the biometric template.

enrolled with the verifying party. The use of ZKPK proof enables us to support *two-factor* (i.e. the BK and the secret random r) verification.

After the online enrollment, all BK-HASH-VECTOR's and the BK are deleted from the individuals machine to prevent information leakage. If enrollment is in-person, the random secret r associated with the BK in BK-COMMIT and the BK-META-DATA are saved by the individual in a portable device that the individual carries.

At the time of verification the individual needs both to provide r and to reconstruct the BK to prove knowledge of the value committed at enrollment. Notice that r is under the individual's control and is never revealed. To re-generate the key, as for the enrollment, the individual re-executes the same process illustrated in Figure 4.1. First the individual provides its biometric reading, which results in the BK-HASH-VECTOR. The BK-HASH-VECTOR is then classified as BK-CLASS-COMB using the BK-META-DATA provided by the individual. The BK-CLASS-COMB is subsequently provided to the cryptographic algorithm to create a valid proof of knowledge which is required for the verification.

Finally, to revoke BK, the BK-COMMIT corresponding to enrolled biometric is to be added to a revocation list which is similar to the certificate revocation list (CRL) [50] in a public key infrastructure. CRL is typically a list of certificate serial numbers that have been revoked, and should not be relied upon by any system entity. In our system, the revocation list consists of the BK-COMMIT's that have been revoked. After publishing the BK-COMMIT in the CRL list, the individual cannot do a proof of knowledge with that BK because it relies on a revoked commitment. Other revocation mechanisms for SIT identifiers can also be used for BK-COMMIT (See Chapter 2 Section 2.5).

4.2 Biometric Key Generation Protocol

In this section we first provide some preliminary concepts related to the main techniques underlying our proposed solution. Then, we discuss the two main algorithms that represent the core algorithms for BK generation– the SVD based image hashing algorithm and SVM classification algorithms.

4.2.1 Preliminary Concepts

There are two key concepts related to our biometric key generation technique. First is the Singular Value Decomposition (SVD) which is used for the various transformation functions of the biometric image hashing. Second is the Support Vector Machines (SVM) that is used to classify various hash vectors obtained from different individuals.

Singular Value Decomposition (SVD) The SVD is a well known technique of modern numerical linear algebra, to factorize a *m x n* matrix into a diagonal form. As proven in [85], if A is a real *m-by-n* matrix, the two orthogonal matrices exist:

$$U = [u_1, \ldots, u_m] \in \mathbb{R}^{m \times m} \; and \; V = [v_1, \ldots, v_n] \in \mathbb{R}^{n \times n}$$

such that

$$UAV^T = diag(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m \times n} \qquad\qquad p = min\{m, n\}$$

where V^T is the transpose of matrix V and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$. σ_i's are the *singular values* of A and the vectors u_i and v_i are the ith *left singular vector* and the ith *right singular vector* respectively. $\sigma_i(A)$ denotes the ith largest singular value of A.

The singular values of a matrix A are unique. The SVD exposes the geometric structure of a matrix A. The singular values σ_i's reflect the variations along the corresponding i singular vectors. It can be shown that computation of the right singular vectors and the singular values can be obtained by computing the eigenvectors and eigenvalues of the symmetric matrix $M = A^T A$ where A^T is the transpose matrix of A.

Support Vector Machines (SVM). SVM [28] is a classifier based on statistical learning techniques developed by Vapnik *et al.* [86]. The techniques find optimal hyperplanes to determine the boundaries with the maximal margin separation between every two classes. This is performed among different classes of the training data in a high dimensional feature space. Then additional data, which was not used during the training, is used as test data and can be classified using the separate hyperplanes.

Let $\{x_i, y_i\}$, $1 \leq i \leq L$ be L training data vectors, having the training data item denoted as x_i which has a class label denoted by y_i, where $y_i \in \{-1, +1\}$ for binary classification. Given an input vector x, SVM constructs a classifier of the form

$$f(x) = Sign(\Sigma_{i=1}^{L} \alpha_i y_i K(x_i, x) + b)$$

where $\{\alpha_i\}$ are non-negative Lagrange multipliers each of which corresponds to an example from the training data, b is a bias constant, and $K(\cdot, \cdot)$ is a kernel function satisfying the conditions of Mercer's theorem [87]. Some frequently used kernel functions are the polynomial kernel $K(x_i, x_j) = (x_i \cdot x_j + 1)^d$ and Gaussian Radial Basis Function (RBF) $K(x_i, x_j) = e^{-|x_i - x_j|^2/2\gamma^2}$. There are several approaches to adopting SVMs to classification problems with three or more classes as well.

SVM applies to classification of vectors, or uni-attribute time series. To classify multi-attribute data, which are matrices rather than vectors, the multi-attribute data must be transformed into uni-attribute data or vectors. Therefore we use the combination of the SVD technique with SVM which has been explored in [88–90]. SVD is used to reduce multi-attribute biometric data to feature vectors.

4.2.2 SVD Image Hashing

In this section we describe the generic hashing mechanism that will be a key component to our BK generation protocol. We build on the algorithm introduced in [91] in Algorithms 1 and 2, and describe its main steps (as illustrated in Figure 4.2) in the following.

Pre-processing. As a first step the biometric image may be pre-processed so as to obtain a clear biometric image block I. This stage aims at choosing a clear and well-focused biometric image. Pre-processing provides an effective region in a selected biometric image for subsequent feature extraction. We consider fingerprint images, iris images and face images.

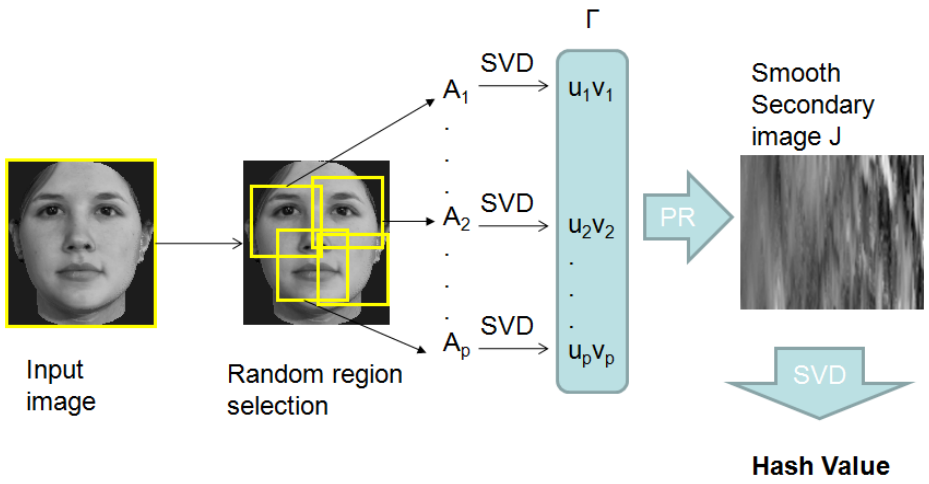


Fig. 4.2. Key steps of the biometric image hashing algorithm.

For the specific case of a fingerprint image, as a part of pre-processing the *region of interest* (ROI) is identified (See step 2 of Algorithm 1). The unique characteristics of the fingerprint are known to be around the core point or delta point [92]. The outside portion of a fingerprint is generally prone to small translations and is typically cropped out. Also, a larger area of the central portion of fingertip skin is in contact with the scanner surface as compared to the peripheries, giving a better image. The center is also better for liveness analysis. This is because data such as the rate of perspiration can be measured because it is higher at the central part of the fingertip as compared to the skin away from the center. Moreover the center region is more robust to pressure dispersion as compared to the other regions. Finally, as the experimental results show, we know that it preserves enough information to identify individuals.

The first step to determine the ROI is to locate the core or delta point. To do this we employ the R92 algorithm of Wegstein [39, 93]. The R92 algorithm begins by analyzing a matrix of angles corresponding to a grid of ridge orientations of the fingerprint which approximately form the trajectories that follow the flow of the ridges. This information

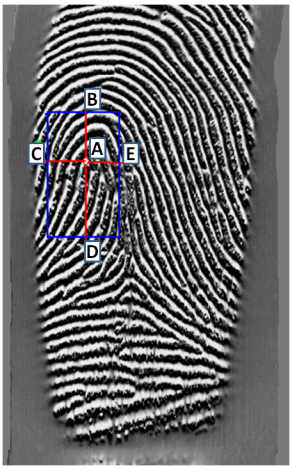


Fig. 4.3. Fingerprint region of interest.

is analyzed to build a “K-table” containing the following values – ⟨ ROW, COL, SCORE, BC SUM, AD SUM, SUM HIGH, SUM LOW ⟩. This table lists the first location in each row of the matrix where the ridge orientation changes from a positive slope to negative slope to form a well-formed arch. This location is captured by the ROW and COL values. Then the SCORE value denoting the nature of a given arch then needs to be evaluated. The point closest to the core or delta point of a fingerprint would have the highest SCORE. If two scores are equal, then as a convention the entry closest to the bottom of the image is considered. To evaluate the SCORE, the nearby arches are analyzed using the variables – BC SUM, AD SUM, SUM HIGH and SUM LOW. The BC SUM is the sum of the arch angle with its east neighbor. The AD SUM is BC SUM plus the one angle to the west and east of the BC SUM. Finally the SUM HIGH and SUM LOW are the summations of groups of angles below the one being analyzed. To calculate these, four angles are combined in a set, and four such sets are individually summed. The SUM HIGH records the highest value and SUM LOW the lowest of these sets. With most entries of the K-table filled, each entry is then scored. Based on the scoring, the core or delta point is identified.

As a next step, the ROI needs to be created. We describe the procedure creating ROI with the help of Figure 4.3. Here the core point is denoted by number 'A'. Considering point 'A' as a benchmark, then four lines are drawn in four orthogonal directions. A variable *count* is initiated at 0. As the length of each line prolongs, if the line encounters a ridge, then the *count* is incremented by one. In Figure 4.3 the four points 'B','C','D', and 'E' are determined when count is 4, which is the threshold size fixed for our experiments.
This threshold was determined based on the experimental results on the fingerprint images captured from Optical and Thermal Sensors [94]. We expect a similar threshold to be valid for other sensors because even if the quality of the fingerprint image may differ, the area of the fingerprint captured is similar.
Based on the four points a rectangle is created which is the ROI. These steps are illustrated from steps 6 to 15 of Algorithm 1. This ROI is then used as an image input for the rest of the algorithm as shown in step 15 of Algorithm 1.

Feature Extraction. Once the image I of size $n \times n$ is finalized, the features are extracted based on random region selection. This is done by choosing p semi-global regions based on a pseudo random (PR) generator that uses a secret key r. The secret key is stored at the user's local machine. The idea of extracting robust feature vectors from PR semi-global regions via matrix invariants has been investigated extensively [95, 96]. A matrix invariant is any function or property of a square matrix which is not altered under a change of basis. More specifically, a matrix invariant is a function or property of the underlying linear transformation, which can be computed from the matrix relative to any basis. The matrices formed corresponding to the selected sub-images are used to be processed under matrix invariant functions such as SVD as elaborated in the transformation step of the algorithm.

The random partitioning of the image introduces unpredictability in the hash values and hence increases the security of the overall system. As long as these sub-images are sufficiently unpredictable, the resulting intermediate hashes are also different with high probability [95]. The squares ρ_i determined in steps 18–23 and used in the partitioning (See Figure 4.2) are deliberately chosen to be overlapping to further reduce the vulnerability

of the algorithm to malicious tampering. Increased number of squares p, increases the pseudorandomness in the resultant hash value, and therefore helps in increased security as explained in Section 4.3, assuming a secure pseudorandom number generator. As a further advantage, the random partitioning decreases the probability of collision and increases the robustness against noise that may be present in the biometric image. As reported in line 22 of Algorithm 1, A_i's where $1 \leq i \leq p$ are matrices corresponding to the selected sub-image blocks. Here each element of the matrix A_i corresponds to the 256 grey level value of the pixel of the selected sub-image. The encoding of the actual matrix used in the transformation is done based on the fact that every element in the matrix has a grey value g where $0 \leq g \leq 255$, position v and direction d. A single pixel may not have a direction, but for a group pixels, the grey value may change hence defining a concrete direction. This is important as isolated significant components may not be robust.

Transformation. Each of the sub-images A_i where $1 \leq i \leq p$, is used to perform the SVD transformation. As a result for each A_i a unitary reduction to the diagonal form is performed to get

$$A_i = U_i S_i V_i^T$$

As such SVD selects the optimal basis vectors in the L_2 norm[3] sense such that, for any $m \times m$ real matrix A_i, we have

$$(\sigma_k, \overrightarrow{u_k}, \overrightarrow{v_k}) = arg\ min_{a,\overrightarrow{x},\overrightarrow{y}} |A - \Sigma_{l=1}^{k-1} \sigma_l \overrightarrow{u_l} \overrightarrow{v_l}^T - a\, \overrightarrow{x}\, \overrightarrow{y}^T|_F^2$$

where $1 \leq k \leq m$, $a \in \mathbb{R}$, $\overrightarrow{x}, \overrightarrow{y} \in \mathbb{R}^m, \sigma_1 \geq \sigma_2 \ldots \geq \sigma_m$ are singular values, $\{\overrightarrow{u_i}\}$ and $\{\overrightarrow{v_i}\}$ (where $1 \leq i \leq p$) are the corresponding singular vectors and $(\cdot)^T$ is the transpose operator [91]. By using the SVD we preserve both the magnitude of the important features in singular values and also their location geometry in the singular vectors. The combination of the left most and right most singular vectors that correspond to the largest singular values, in turn capture the important geometric features in an image in the L_2 norm sense.

[3] L_2 norm, also known as the Euclidean norm, defined for a vector $\overrightarrow{x} = \{x_1, \ldots, x_n\}$ is denoted by $|\overrightarrow{x}| = \sqrt{\sum_{k=1}^{n} |x_k^2|}$.

Therefore as a next step for each A_i, the $\overrightarrow{u_i}$, which is the first left singular vector and the $\overrightarrow{v_i}$, which is the first right singular vector are retrieved. All these vectors are then combined in $\Gamma = \{\overrightarrow{u_1}, \ldots, \overrightarrow{u_p}, \overrightarrow{v_1}, \ldots, \overrightarrow{v_p}\}$.

The next step is to form a PR smooth secondary image J from Γ. J is formed according to an iterative process, at each step of which an element from Γ is selected and added to J. As a first step an element is pseudo randomly selected from Γ and set at the first column of J. Then for the i^{th} column of J, an element from Γ is selected such that it is closest to the $(i-1)^{th}$ column of J in the L_2 norm sense as denoted in step 16 in Algorithm 2. An element can only be chosen once from Γ, therefore an element chosen at the i^{th} step cannot have been chosen at any of the previous $(i-1)^{th}$ steps. Hence after $2p$ steps all the elements of Γ are pseudo-randomly reordered to form the secondary image J of size $m \times 2p$.

Once J is formed SVD is re-applied to it, to finally obtain the image hash vector (steps 26 – 29 of Algorithm 2). The left and right singular vectors are obtained by $J = U_J S_J V_J^T$. Then the singular vectors corresponding to the largest singular values, that is, the first left ($\overrightarrow{u_J}$) and first right ($\overrightarrow{v_J}$) are chosen. These vectors are simply combined to obtain the final hash value $\overrightarrow{H} = \{\overrightarrow{u_J}, \overrightarrow{v_J}\}$.

4.2.3 SVM Classification

As discussed in the previous section, from one input biometric sample, a hash vector $\overrightarrow{H} = \{\overrightarrow{u_J}, \overrightarrow{v_J}\}$ of size $m + 2p$ is obtained. As the hash vectors obtained from different biometric samples of the same user may be the same or may differ from sample to sample; we train a classifier to determine which hash values correspond to a given user (or class), so that at the time of verification, the classifier can identify the correct class of the user. To achieve this goal several biometric samples of different users are taken. Algorithm 1 and 2 are run on each sample to get the corresponding hash vector.

These samples are then divided into training and test data to perform the classification. We use K-fold cross-validation to divide the training and testing data. All sample hash

Algorithm 1 Generic Biometric Image Hashing Algorithm

Require: Biometric image I

Ensure: The quality of the image is suitable based on the biometric.

Input biometric image I
{**Pre-process fingerprint images to calculate ROI**}
if (type(I) == 'fingerprint') **then**
  $point_1$ = Algorithm_R92(I) {Compute the core or delta point}
  size = 4 {Set fingerprint ROI threshold size}
  count = 0
  for each line i in the four orthogonal directions (N,S, E, W) **do**
    repeat
      increment length of line;
      if line encounters a ridge **then**
        $point_i$ = coordinate of intersection of line and ridge
        count++
      end if
    until (count $\neq$size)
  end for
  I = crop($point_2, point_3, point_4, point_5$)
end if
Let resultant image $I \in \mathbb{R}^{n \times n}$ be of size $n \times n$
{**Random Selection**}
Let p be the number of rectangles
Let ρ_i be the i^{th} rectangle and m be the height and width of each ρ_i.
for each i where $1 < i < p$ **do**
  Randomly position rectangle ρ_i at (x_i, y_i) such that $x_i + m < n$ and $y_i + m < n$
  Let A_i be the "sub-image" that is formed by taking the portion of image that is in ρ_i : $A_i \in \mathbb{R}^{m \times m}, 1 \leq i \leq p$.
end for

Algorithm 2 Generic Biometric Image Hashing Algorithm (Continued)

(Continued from Algorithm 1) {**First SVD Transformation**}
for each A_i where $1 \leq i \leq p$ **do**
  $A_i = U_i S_i V_i^T$ {Collect singular vectors corresponding to the largest singular value}
  $\overrightarrow{u_i}$ = first left singular vector
  $\overrightarrow{v_i}$ = first right singular vector
end for
$\Gamma = \{\overrightarrow{u_1}, \ldots, \overrightarrow{u_p}, \overrightarrow{v_1}, \ldots, \overrightarrow{v_p}\}$
Initialize secondary image $J[m, 2p]$ {Constructing secondary image from singular vectors}
for all c where $1 \leq c \leq 2p$ **do**
  Initialize variable e_c corresponding to element in Γ
  if $c = 1$ **then**
    $e_c = PR\_Select(\Gamma)$
  else
    var_loop = true
    while var_loop **do**
      $e_c = min_{k=1}^{2p}(\sqrt{\sum_{l=1}^{c-1}(J(l) - \Gamma(k))^2})$
      if not(e_c already chosen for J) **then**
        var_loop=false
      end if
    end while
  end if
  for all r where $1 \leq r \leq m$ **do**
    $J[r][c] = e_c[r]$
  end for
end for
{**Second SVD Transform**}
$J = U_J S_J V_J^T$ {Collect singular vectors corresponding to the largest singular value}
$\overrightarrow{u_J}$ = first left singular vector
$\overrightarrow{v_J}$ = first right singular vector
$\overrightarrow{H} = \{\overrightarrow{u_J}, \overrightarrow{v_J}\}$
return Hash Value $\overrightarrow{H}$

vectors are partitioned into K subsamples. Of the K subsamples, a single subsample is retained as the validation data for testing the model, and the remaining K - 1 subsamples are used as training data. The cross-validation process is then repeated K times (the folds), with each of the K subsamples used exactly once as the validation data. The K results from the folds are then averaged to produce a single estimation [97].

The obtained hash vectors do not differ greatly with respect to Euclidean distance, as inferred through experimental analysis therefore we use SVM techniques to map the input hash vectors onto a higher dimensional space where a maximal separating hyperplane can be constructed.

As explained in Section 4.2.1 the hyperplane constructed using SVM is such that it has maximum distance to the closest points of the training set. These closest points in the training set are called *support vectors*. Here we use the Gaussian radial basis kernel function (RBF for brevity) $K(\overrightarrow{H}_i, \overrightarrow{H}_j) = e^{-|\overrightarrow{H}_i - \overrightarrow{H}_j|^2/2\gamma^2}$ where $\overrightarrow{H}_i$ and $\overrightarrow{H}_j$ are two of the training samples and $\gamma > 0$.

During training, two specific parameters have to be assessed, namely γ used in the RBF kernel function and the penalty parameter C that is used in the evaluation of an optimal hyperplane balancing the tradeoff between error and margin. We perform a grid search method [98] on the parameters C and γ try all combinations of C and γ to selected the pair with the best CV accuracy.

After training, the SVM model encodes all the classes that this SVM classifier has been trained with. If one of the classes were to represent the final BK, then the SVM model itself would reveal significant information about the BK. The adversary would have to simply guess the right (single) class to guess the right BK. We mitigate such threat by strategically adding spurious classes in the model and by generating the final BK as a combination of different classes as described in the following.

Adding Spurious Classes

Additional classes can be added to the original SVM classifier model by training additional samples of the given biometric. These samples have to be added so as they are indistinguishable with respect to the original biometric classes. We therefore employ a strategy to make the additional classes similar to the original set of classes. For each class in the SVM model we define a *protector class* that is similar to the original class so that the cluster formed by the *protector class* is close to the original SVM class, and yet is different enough to be distinguished as a different class. There could be different ways of obtaining the protector classes. First is to find biometric images of different individuals that look perceptually similar. In the case of fingerprints perceptually similar images may be easy to find, however in the case of face images comparing similar face images requires a human to evaluate the closeness. A second possibility is to add noise to the original biometric image. For example the face images could be modified to render naturally asymmetric features to symmetric or changing other specific aspects as the size of the face characteristic such as the eyes, nose and so on. If there are n original classes, then we manually add a protector class for each, thus resulting in $2n$ classes. We also add other spurious classes that are not similar to the original biometric samples (as the protector classes) but are of the same biometric type. Adding spurious classes by itself does not secure against brute force guessing attacks, but increasing the resultant number of classes in the classifier also increases the computational hardness of the guessing attacks on the BK value (See Section 4.3). The BK value is the combination of the SVM classes as described next.

Using Combination of Classes

During classification, given an input sample, a SVM classifier returns a list of confidence measures for each class corresponding to the prediction of the class of the input sample. Generally the class with the highest confidence is taken as the overall class prediction of the input sample. The distance from the separating hyperplane and other multiple factors are used in determining the confidence of a prediction per class [99]. Using the con-

fidence factors, not only is the final prediction known, but also a list of other classes that have high confidence levels. While generating the final key, a combination of the classes that are chosen based on the class prediction confidence of the SVM classifier is used. More specifically if n is the total number of classes, then the final BK is the label of class with the highest confidence label and an unordered combination of the top $t = \frac{n}{2}$ class labels that are listed with decreasing confidence levels. For an attacker to guess the BK-COMB-CLASS, given the SVM classes, the number of choices is $n + \binom{n}{t}$. We typically consider the total number of classes $n > 69$, which leads the number of choices to be $> 2^{64}$, thus making it computationally hard for the attacker to guess the right BK-COMB-CLASS.

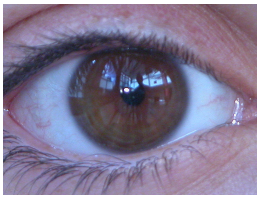
4.2.4 Experiments

In this section we summarize the experimental results we conducted to assess the accuracy and robustness of our approach. We carried out extensive sets of tests of different biometrics, to demonstrate that the relevant criteria required for the security, repeatability and uniqueness of the BK are met. All experiments have been conducted using Microsoft Windows XP Professional 2002 Service pack 1 operating system, with Intel(R) Pentium(R)4 3.20GHz and memory of 512MB.

Dataset and Experimental Setup

We tested our hashing algorithm (Algorithms 1 and 2, Section 4.2.2) on fingerprint, iris and face data. The summary of the data used and the obtained results are reported in Table 4.2. For fingerprints we used FVC [94] databases (See Figure 4.4(a)). The FVC dataset used consists of 324 fingerprint images of 59 individuals collected using thermal sweeping and optical sensors. We also used 50 images of 10 individuals generated using the synthetic fingerprint generator SFingeGe v3.0 [100]. Regarding the iris data, the UBIRIS iris Database3 [101] was used (See Figure 4.4(b)), which consists of 1695 images of 339 individuals' eyes. Finally for the face data we used the Yale Database of Faces [102] containing 100 images of 10 individuals (See Figure 4.5(a)) and the AT&T

(a) Thermal (left) and optical (right) sensor fingerprint samples.



(b) Iris sample.

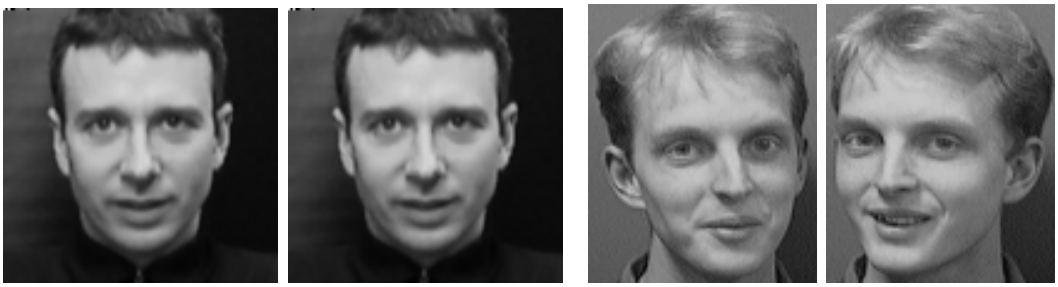
Fig. 4.4. Fingerprint and iris image samples.

Database of Faces [103, 104] (See Figure 4.5(b)) containing 400 images of 40 individuals. We evaluated our results using the SVM classification algorithm, with K-fold cross validation (CV). Based on the CV accuracy the False Acceptance Rate (FAR) and False Reject Rate (FRR) were calculated. The FRR is calculated as $1 - CV\ Accuracy$, whereas the FAR is calculated as the $\frac{FRR}{N-1}$ where N is the number of classes.

The values of the key parameters used in Algorithms 1 and 2 are reported in Table 4.1 where n is the size of the image in pixels, p is the number of sub-images formed, m is the size in pixels for each of the sub-images and J is the secondary image.

To assess the optimal values of the pair p and m, we ran various possible combinations of the values and used the one that provided the maximum accuracy. For example for fingerprint database FVC2004 DB3_B, the value of p was varied between $[10, \ldots, 100]$ and that of m also varied between $[10, \ldots, 100]$, the highest accuracy was found for $p = 50$ and $m = 30$.

The code for implementing the various steps is written in MATLAB and the `rand()` function of MATLAB is used as the pseudo random function used in step 21 and 12 of



(a) Yale face database samples. (b) ATT face database samples.

Fig. 4.5. Face image samples.

Algorithms 1 and 2 respectively. The size of the secondary image J is 30×100 leading to the size of $\overrightarrow{u_J} = 30 \times 1$ and $\overrightarrow{v_J} = 100 \times 1$, thus resulting in a hash vector $\overrightarrow{H} = \{\overrightarrow{u_J}, \overrightarrow{v_J}\}$ of 130 dimensions.

For the SVM classification we adopted the LIBSVM [98] package to use the hash vectors and build the final classifier model. This uses the RBF as the kernel function. Based on experimental analysis, C was set to the range $\{2^5, \ldots, 2^{15}\}$ and γ to $\{2^{-5}, \ldots, 2^3\}$. All combinations C and γ were tried using grid search to select this best CV accuracy based on the input data.

Table 4.1 Parameter values for experiments based on Algorithms 1 and 2.

Image type	n	p	m	size of J	pseudo-random function	size of $\overrightarrow{H}$
Fingerprint/Iris/Face	128	50	30	30×100	MATLAB `rand()`	130

Table 4.2 Summary of the experimental results of all biometric data types.

#	Biometric Type	Database Name	Description	# Images	# Persons	CV Accuracy %	FRR %	FAR %
1.	Fingerprint	FVC2004, DB3_B	300×480, Thermal Sweeping Sensor	54	9	92.59	7.41	9.26 $\times 10^{-03}$
2.	Fingerprint	FVC2004, DB3_A	300×480, Thermal Sweeping Sensor	150	30	97.33	2.67	9.21 $\times 10^{-04}$
3.	Fingerprint	FVC2004, DB2	328×364, Optical Sensor	120	20	85.83	14.17	7.46 $\times 10^{-03}$
4.	Fingerprint	SFingGe v3.0, Synthetic Generator	288×384	50	10	88	12	1.33 $\times 10^{-02}$
5.	Iris	UBIRIS.v1 Sessao_1	$800 \times 600 -$ 24 bit color	1100	220	87.73	12.27	5.6 $\times 10^{-04}$
6.	Iris	UBIRIS.v1 Sessao_2	$800 \times 600 -$ 24 bit color	595	119	97.65	2.35	1.99 $\times 10^{-04}$
7.	Face	The Yale Face Database B	$640 \times 480 -$ 8 bit gray scale	100	10	99	1	1.11 $\times 10^{-03}$
8.	Face	AT & T Databases of Faces	$92 \times 112 -$ 256 bit gray scale	400	40	98.25	1.75	4.49 $\times 10^{-04}$

4.2.5 Experimental Results

We now present the results of our experimental evaluation of our system. First, regarding the time performance, on an average, the hash vector from any given image is generated in 0.9597 seconds. In the process of enrollment if we need 5 samples of one person, it will take about 5 seconds of processing time to generate hash vectors for one person for the given software and hardware setting.

The SVM classifier for about 220 persons' hash vectors takes 3 or 4 hours to generate the SVM classifier model. At the testing stage, when the model is generated, it takes approximately 0.001 second to classify the test image.

Regarding the experimental results, overall, the obtained results largely confirm the correctness of our algorithm: in each of the test cases, the accuracy was above 85% cross validation. False acceptance rates were low ranging between $[1.99 \times 10^{-04}, \ldots, 1.33 \times 10^{-02}]$, which translates into the assurance that the percent chances of accepting an incorrect biometric image are low. The worst FAR has been registered as 1.33×10^{-2}, which is interestingly obtained from the synthetic fingerprint images. These images are generated under controlled conditions (e g., there is no unexpected noise caused by human interaction). Regarding FRR, the worst recorded FRR was in conjunction with the worst accuracy results because the FRR result is dependent on the accuracy (see previous section). The worst rate amounts at 14% (case test n. 3) and it is still acceptable, as it is in the same order of similar biometric key generators [105].

We now provide additional insights specific to the tested biometric types.

Fingerprint. Two types of Fingerprint Verification Competition (FVC) databases [94] corresponding to two types of sensors were used for the fingerprint biometric experiments. The sensors highly influence the quality of fingerprint images. We define the *quality* of the fingerprint image according to three criteria [73]: (i) high contrast between ridges and valleys, (ii) the image area foreground, and (iii) little scar or latency. As shown by the results, the CV cross validation is above 85% for each data set considered, which confirms the validity of our approach. A first important consideration suggested by the result is that

the algorithm performs better in case of a large data set (as in the test case n. 2 in Table 4.2), most likely because of the more accurate training and testing during configuration phase which helped in finding the optimal configuration parameters. We also notice that on average our algorithm performs better using the thermal sensor than when using the optical sensor because the thermal sensor captures better quality fingerprint images. We can explain this result by elaborating more on how the quality is affected; in that the quality of the fingerprint image is affected by several human factors such as skin humidity and pressure. If the skin humidity is less the image quality of the optical sensor degrades. The skin humidity does not affect the image quality of the thermal sensor because it is the sweeping type. Moreover, regarding pressure, for an optical sensor the foreground image gets smaller for low pressure while the fingerprint is smeared for high pressure. This is again not true for a thermal sweeping sensor where the image quality is not affected significantly.

Out of the four data sets the last one was composed of artificially generated images. We experimented with synthetic fingerprint images as they potentially supply non-biased images and can be created at a low cost. However, it was difficult to control the randomness, which led to significant differences in the fingerprints generated for the same individual. This caused a degradation in the classification to get a 88% cross validation accuracy. We believe the results could be improved using a synthetic generator version that generates several fingerprint samples maintaining some known invariant fingerprint features as in the case of real fingerprints. This would guarantee random noise for each sample, as in the case of real life sensors, while keeping the basic characteristics, such as the nature of the core point (whorl versus delta), unchanged.

Iris. Iris image databases UBRIS.v1 Sessao_1 (Session 1) and UBRIS.v1 Sessao_2 (Session 2) [101, 106] were used for the iris biometric experiments. For the first image capture session, the noise factors such as reflections, luminosity and contrast were minimized. In the second session the capture place was changed to introduce natural luminosity factor. Images collected in the second session simulated the ones captured by a vision system without or with minimal active participation from the subjects, adding possible noise in the

resultant images. When capturing iris images there is a certain amount of pre-processing performed. A sequence of images are obtained rather than a single image. Not all images in the input sequence are clear and sharp enough for recognition. The images may be out of focus, or contain interlacing lines caused by eye motion or have severe occlusions caused by eyelids and eyelashes. Therefore, only high quality images from an input sequence are included in the final database.

The number of classes in the first database was almost double of the second, therefore the increased number of hyperplanes formed during SVM modeling led to an increased number of misclassifications. This resulted in better results for the second database, even if the quality of images of the first database was better. Therefore we observe that the increased number of classes (> 200) in the SVM classification model influence the final cross validation accuracy.

Face. We used two databases for the experiments on faces. The first one collected good quality images, in that photos were taken with each subject in a frontal pose (See Figure 4.5(a)). Thus the resulting cross validation accuracy of 99% was obtained.

The second set of tests were performed on images that were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position with tolerance for some side movement (See Figure 4.5(b)). Despite this, the overall cross validation accuracy of this database was evaluated to be 98.25% although the false rejection rate increased by .75%.

4.3 Analysis

We start with proving some key properties related to uniqueness and repeatability and security properties of the biometric key algorithms. Based on such results we analyze the privacy aspects, the possible attacks and discuss how to prevent from them.

4.3.1 Uniqueness and Repeatability

Uniqueness and repeatability are two key requirements for biometric keys as mentioned in the beginning of this chapter. We analyze these two properties in detail in this section.

A criterion frequently used for assessing uniqueness and repeatability in classification, is the function J_2 [107] which is described in the following. The key idea of function J_2 is to compare the *within-class* distance of the various hash vectors (or elements being classified) belonging to a given class, with the *between-class* distance among the various classes. There are two key steps to be taken while evaluating J_2.

The first step is to evaluate the *within-class* scatter matrix S_w:

$$S_w = \Sigma_{i=1}^{M} S_i P_i$$

where M is the total number of classes; $S_i = E[(x-\mu_i)(x-\mu_i)^T]$ is the covariance matrix[4] for a class denoted by w_i where E is the expected value function, x is any vector in class w_i and μ_i is the mean vector of class w_i; and, $P_i = n_i/N$ where n_i is the number of samples in class w_i and N is the total number of samples in all the classes.

The second step is to evaluate the *between-class* scatter matrix S_b:

$$S_b = \Sigma_{i=1}^{M} P_i(\mu_i - \mu_o)(\mu_i - \mu_o)^T$$

where $\mu_o = \Sigma_{i=1}^{M} P_i \mu_i$ is the global mean vector of all the classes.

From the above a covariance matrix of feature vectors with respect to the global mean is evaluated as $S_m = S_w + S_b$. Finally the J_2 criterion is calculated as:

$$J_2 = \frac{|S_m|}{|S_w|}$$

[4]Covariance is the measure of how much two random variables vary together. A covariance matrix is a matrix of covariances between elements of a vector.

As evident from the equation, for good repeatability of correct classification (small within-class distance), and uniqueness (large between-class distance) the value of J_2 should be large.

We carried out additional experiments on all the datasets to estimate J_2 and obtained average values of J_2 for fingerprint as 1.2712×10^{81}, iris as 1.5242×10^{303} and face as 3.7389^{103}. This provides empirical evidence that the algorithm satisfies the uniqueness requirement on the biometric hashes generated based on the biometric datasets we used.

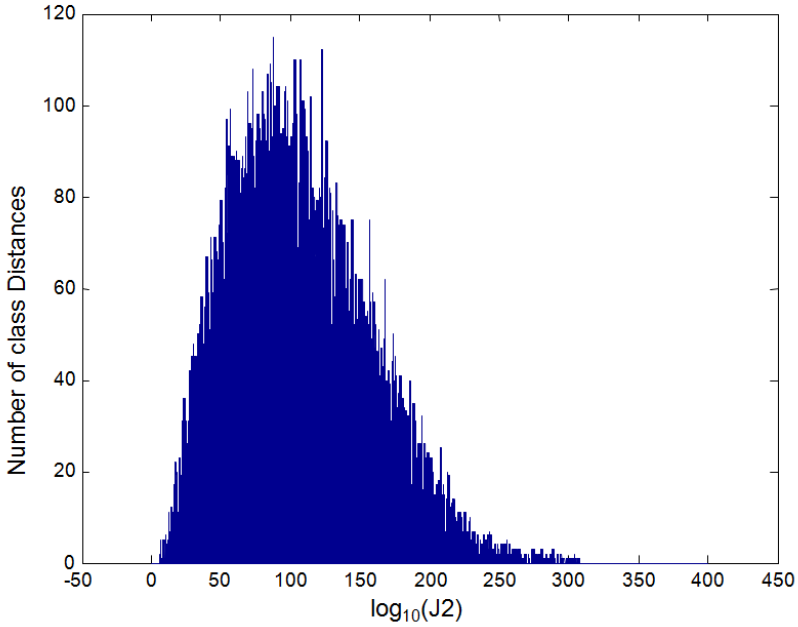


Fig. 4.6. J2 histogram of iris classification.

For clarity, we provide an example of a J_2 histogram for the Iris Session 1 database in Figure 4.6 (data corresponding to n. 5 in Table 4.2). The J_2 metric requires the calculation of *within class* and *between class* distances of all the possible pairs of data elements. The y axis in the histogram presents number of log(J2) class distances between any two classes.

For instance for a value (120(x-axis),100(y-axis)) means that there are 100 class distances which have the J_2 value of 120. If there are all together $|C|$ number of total classes then the possible permutations of the distances to be tested are $\frac{|C|\times|C-1|}{2}$. In Figure 4.6 which the histogram of different classes of iris data there are 220 individuals iris data therefore the possible number of distances between any two classes is equal to $\frac{220\times119}{2}$. This results in the projected histogram.

4.3.2 Biometric Image Keyed Hashing

We analyze the one-way security property of the SVD based biometric image hashing algorithm. More specifically, we show that it is computationally hard, given BK-HASH-VECTOR $\overrightarrow{H}$ to reconstruct the original biometric image. We explain this result by the following two theorems. First, we prove that it is hard to construct the secondary image from the vector, which is required to reconstruct the original biometric. The following result (Theorem 4.3.2) shows that even if the second image is constructed or attacked, it is still hard to obtain the original biometric image I. Our proofs are based on the combination of mathematical properties of the SVD and the employed hashing technique, and from empirical analysis where image-specific properties need to be analyzed. Experimental results are required to define important thresholds and boundaries for biometric specific data that cannot be proven theoretically.

Theorem 4.3.1 *Given only the hash value $H(u_J, v_J)$ it is computationally hard to construct the secondary image.*

Proof If only the final hash value is known to an adversary, then the first step is to approximate the secondary image J (See Figure 4.2). We prove it by analyzing the following equation which provides a possible approximation of the secondary images:

$$J = \sum_{i=1}^{r} \sqrt{\lambda_i} u_i v_i^T = \sqrt{\lambda_1} u_J v_J^T + \underbrace{\sqrt{\lambda_2} u_2 v_2^T + \sqrt{\lambda_3} u_3 v_3^T + \ldots + \sqrt{\lambda_r} u_r v_r^T}$$

Here $r = 2p$, p is the number of sub-images created, λ_i, $1 \leq i \leq r$ are non-zero eigenvalues of the matrix $J^T J$ such that $\lambda_1 > \lambda_2 > \ldots > \lambda_r$. Note J^T is the transpose matrix of J and a positive square root of λ_i is a singular value. The u_i and v_i are eigenvectors of JJ^T and $J^T J$ respectively. As the final hash value [u_J,v_J] is known to the adversary, the values which need to be guessed are λ_1 and $\{\lambda_2 u_1 v_1^T + \lambda_3 u_2 v_2^T + \ldots + \lambda_r u_r v_r^T\}$. To guess λ_i's there are infinitely many solutions as any nonnegative eigenvalues can lead to specific eigenvectors that are unitary (i.e. satisfy the definition). Any eigenvalue matrix resulting from this construction will give a solution to the equation and therefore it is computationally hard for the adversary to identify the original value.

If there is a case when the λ_1 is dominant such that the rest of the values $\lambda_2, \ldots, \lambda_r$ are approximately equal to zero, then one could try to guess the λ_1 and possibly approximate the secondary image to get $\dot{J} = \sqrt{\lambda_1} u_J v_J^T$. It is not trivial to predict theoretically the possible distribution of the values of λ_i's because they are dependent on the type of image and the distribution of the pixel values of those images. Therefore we conducted experimental evaluation on the biometric images and found that the λ_i's are distributed such that there is no one dominant eigenvalue because the secondary image J is a smooth image (i.e. the adjacent pixels of the image do not differ beyond a certain threshold which is determined by the algorithm parameters). We conclude that because of the difficulty of guessing the eigenvalues and the lack of dominant eigenvalues the reconstruction of the secondary image J from the resultant hash vector $\overrightarrow{H}$ is computationally hard for the biometric images considered. ■

Theorem 4.3.2 *Given the secondary image it is computationally hard to get the original image I.*

Proof Sketch If J is known to the adversary, then the first step would be to form each sub-image matrix A_i where $1 \leq i \leq p$. Note a combination of all A_i eigenvectors were used to construct J. Each A_i is of the form $A_i = U_i S_i V_i^T$. As in the proof of Theorem 4.3.1, there exist infinite number of eigenvalues to construct infinite A_i which would satisfy the relation. Moreover, using the same reasoning as before, there are no dominant eigenvalues

as the p sub-images each of size $m \times m$ are overlapping. The overlap causes the most significant eigenvalues not to differ beyond a certain threshold as determined by the algorithm parameters p and m. In addition the largest eigenvectors (i.e. the left most and the right most vectors of the U_i and V_i matrices respectively) of each sub-image A_i are combined pseudorandomly to form J resulting in the number of choices the attacker would need to try as $p!$. This motivates the need for large values of p (~ 50). As a result guessing the order of each sub-image A_i and hence creating the original image I is computationally hard. ■

As a final remark we note that even if the attacker is able to retrieve the biometric image, it cannot reconstruct the hash vector without the knowledge of the secret random value needed during the selection of the p sub-images and to pseudorandomly combine them to form the secondary image J.

4.3.3 SVM Classes and Key Space

We analyze how the addition of spurious classes in the model affects the security of the final BK.

Spurious Classes

We performed empirical analysis to evaluate how the classification accuracy changes as we add the spurious classes. We carefully selected spurious classes so they would not be similar to the biometric samples already present in the classifier but are of the same biometric type. We also processed protector class images for each class in the original classifier model. In Figure 4.7 we show three different cases used for biometric images of the face database. Here there are $n = 20$ original set of classes. In the first case (dashed line) 10 dissimilar spurious classes added to the original set of n classes resulting in the maximum of $n + 10$ classes. The value in the x-axis ranging from $[1, \ldots, n]$ dictates the number of classes that are trained and tested at one instance, and the value in the y-axis provides the classification accuracy of those number of classes. The accuracy decreases gradually only up to 98.33% as the number of neighboring classes increased up to $n + 10$.

In the second case (dash-dot line) only the n protector classes are added per each of the n original classes, giving a total of $2n$ maximum number of classes. As in the previous experiment we registered a gradual drop, with a final accuracy of 98.25% which is close to the original. Finally in the third case (solid line) we added both the protector classes and the other spurious classes resulting in the maximum of $2n+10$ classes, and obtained a final value of 96% accuracy. Based on the three cases, we conclude that spurious classes can be added strategically without changing the original accuracy by a significant amount ($< 3\%$ in the given dataset).

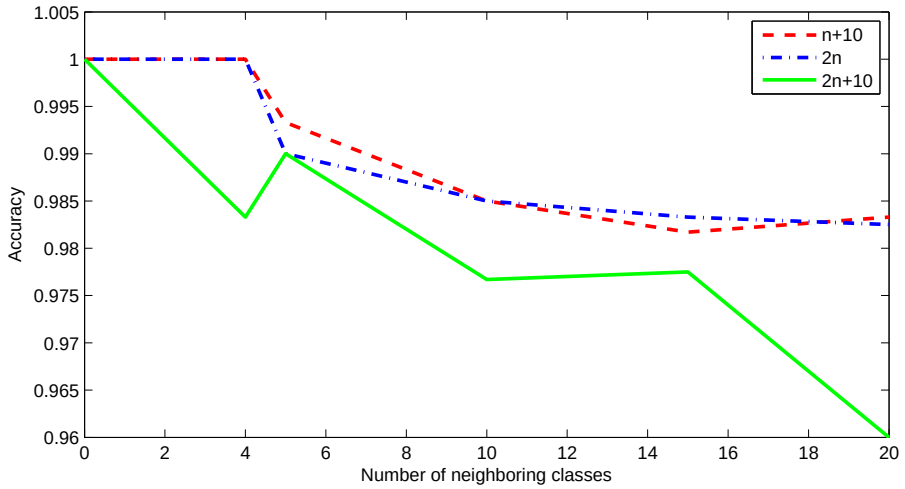


Fig. 4.7. ROC curve showing the affect of spurious classes on the accuracy.

Combination of Classes for final BK

From empirical analysis we observe that if n is the total number of classes, and they are listed in decreasing order of their confidence level, then for instances of correct classification, the highest confidence class is the same and the unordered set of the following t classes where $(n-1) \geq t \geq \frac{n}{2}$ are the same for the multiple testing rounds in the K-fold

validation. The ordering of several of the t classes were swapped with the neighboring classes in several instances. Therefore for the final label that denoted the final BK value, we use the class with the highest confidence followed by an unordered combination of the next t classes. Thus for an attacker to guess the right key based on the classifier model, the number of choices would be $\eta = n + \binom{n}{t}$. n in our case ranges from $[69, \ldots, 220]$. Based on the value of n, the resultant η ranges from $[2^{64}, \ldots, 2^{214}]$. The η is proportional to the number of bits needed to encode the BK. More precisely the number of bits is $\log_2 \eta$ which thus ranges from $[64, \ldots, 214]$. A summary of the experimental data corresponding to the biometric type, n, η and final number of bits of the BK is provided in Table 4.3.

Table 4.3 Number of SVM classes and the final number of bits of the biometric key.

Type	n	Spurious classes	η	# of BK bits
Fingerprint	69	-	2.84×10^{19}	64
Fingerprint	139	69+1	2.36×10^{40}	134
Iris	220	-	4.52×10^{64}	214
Iris	119	-	2.43×10^{34}	114
Face	101	50+1	1.01×10^{29}	96

4.3.4 Privacy and Security Analysis

We analyze the relevant privacy and security properties of our system, based on the above results.

Privacy Analysis Our approach supports privacy preserving verification because it relies on the ZKPK protocol. This ensures minimal disclosure and unlinkability as per the use of the BK identifiers in the aggregate ZKPK as elaborated in Chapter 3. Moreover, with regards to the privacy of the biometric we do not require the storage of the biometric image.

This is a feature of our technique and differs from traditional biometric approaches [108] that rely on matching of the biometric templates. We do not base the matching of the biometric on stored templates, but on hashing values obtained after processing the biometric images. As such, the verifier will not learn anything about the actual biometric image used. Even if a malicious verifier has the BK it is not possible for it to reconstruct or retrieve the original biometric image, which follows from Theorem 4.3.1 and 4.3.2.

Preservation of privacy also prevents replay attacks in supervised environments, which corresponds to the process under the control of the verifier. To succeed in a replay attack the attacker should be able to input a biometric image (in addition to other secrets) to re-generate the BK. If the verification using the biometric is executed in a supervised environment the mere knowledge of the BK would not be sufficient in passing the verification process.

Security Analysis Security in our system is given by difficulty of perpetrating impersonation attacks and of learning additional information about the individual's biometric based on the biometric keys. We discuss possible attacks that an attacker may attempt against our verification system. We focus on an attacker trying to impersonate a different user and show how our approach withstands these types of attacks.

To succeed in an impersonation attack the attacker needs to be aware of all the secrets, and/or bypass the verification methods and compromise the others. Bypassing the cryptographic protocol is computationally hard, as shown in Section 3.4 of Chapter 3. By contrast, it is fairly easy to obtain the biometric image of a legitimate user, because of the image availability[5]. An attacker can present fake biometrics, resubmit previously stored biometric images, override the image extraction process and so forth. As the reliability of the biometric devices and their security cannot be always ensured, it is crucial to verify that our system is indeed capable of thwarting from the attacker having a biometric image[6].

[5]The reader can think of several technologies that make it easy to capture biometrics such as face images.
[6]From our perspective we do not verify whether the biometric image being read by the sensor is synthetic or real.

Table 4.4 Possible security attacks [Key: (a) biometric image (b) hashing secrets (c) classifier model (d) BK (e) commitment secret].

Case	(a)	(b)	(c)	(d)	(e)	**Description**
1	$\times$					The BK cannot be created without the hashing secrets.
2	$\times$	$\times$				The lack of classifier model prevents from the construction of BK.
3	$\times$		$\times$			Classifier model does not reveal secrets and the hashing secret is needed to construct BK-HASH-VECTOR to classify into BK.
4	$\times$	$\times$	$\times$	$\rightarrow \times$		The BK is compromised, but the commitment secret prevents from creating ZKPK.
5				$\not\leftarrow \times$		The BK is compromised, but the commitment secret prevents from creating ZKPK. No other secrets are leaked.
6		$\not\leftarrow \times$	$\times$	$\times$	$\times$	The BK is compromised and so the verification. However, no replay in supervised environment is possible.
7		$\times$	$\times$		$\times$	The client machine is compromised but the BK cannot be constructed without the biometric image.

In Table 4.4 we provide a summary of the various cases where one or more secrets are compromised, and discuss possible security implications. Case 1, 2 and 3 address the scenarios where the biometric image is known to the attacker, but not the BK-METADATA, which includes the hashing secret and classifier model, nor the random secret in the BK-COMMIT, which are stored by the user. Thus, the BK in these cases cannot be generated. However, if the attacker knows the BK, then to perform successful verification it also needs the commitment secrets. This scenario is summarized by case 4. As noted earlier the knowledge of BK does not reveal any information of the biometric image or the secrets involved as depicted in case 5. Moreover, in case 6, if a BK is compromised along

with the BK-META-DATA and the commitment secret, then as highlighted in our privacy analysis, if the verification process is in a supervised environment, then it will not succeed.

Finally, an interesting case is when the user machine storing the BK-META-DATA and the commitment secret are compromised as illustrated in case 7. In this case, the attacker's best choice as a source of information is the SVM model. However as we show in Section 4.3.3, for number of classes $n > 69$, the number of choices $> 2^{64}$ that makes it computationally difficult for the attacker to guess the right BK.

Multi-factor Approach. Considering a multi-factor verification approach presented in Chapters 2 and 3, where multiple identifiers are provided to the verifier, having one of more biometrics as additional factors increases the robustness of the system, as a consequence of the above results. The multi-factor verification using multiple SIT attributes corresponding to users different strong identifiers, stored in the identity record can be used with the BK commitments, which would also be included in the identity record, and subsequently be aggregated together to construct ZKPKs as per the verification criteria of the verifiers. Details on the analysis of the multi-factor approach follow from Chapter 3, Section 3.4.

4.4 Summary

In this chapter we presented a new way to generate BKs from 2D biometric images. These BKs can be used together with other SIT attributes using the multi-factor identity verification techniques presented in Chapter 3. Our algorithms are based on image hashing functions and support vector classification techniques. Through empirical analysis we show that the algorithms provide unique and repeatable BKs for the given dataset. Overall our evaluation uses 2569 images of 488 different individuals for three types of biometric images; namely fingerprint images, iris images and face images. Based on the biometric type and the classification models, we can generate keys ranging from 64 bits up to 214 bits.

We also ensure security and privacy of the biometric data. More specifically, we analyze attack scenarios including the case when all data stored at the client machine is compro-

mised; even in this case the biometric key is not compromised. We preserve privacy of the biometric, in that no information about the original biometric image is revealed from the biometric key or commitment.

There are specific assumptions to consider while employing the biometric key generation and verification protocols. First, our approach assumes that the client has a portable storage device that contains the secrets needed to re-generate the biometric keys. Second, the computational device (either at the client or verifier) that captures the biometric and computes the BK is assumed to be trusted to delete the biometric data and related secrets used in the computation. The computational device is also assumed to be trusted to not release the secret keys during the BK proof computation.

The confidence on the linkability of a biometric commitment and a real world individual can be evaluated based on the type of enrollment and verification policies. Our techniques are designed to preserve privacy and not reveal biometric features or other uniquely identifying information based on the BKs or the proof constructed at verification. In addition an individual can generate multiple BKs using the same biometric, by employing different hashing and commitment secrets. However, if our techniques were to be used for unique identification of individuals then the enrollment and verification mechanisms would need to combine the biometric commitments with other strong identifiers at enrollment and/or verification.

The biometric keys generated can potentially be used for various types of applications such as encryption or other challenge-response based verification. However, based on the applications and the desired security and privacy properties, the algorithms need to be re-assessed with respect to the use of the key generation secrets, constraints on the computational devices and the properties of the final key generated. Examples of the properties of the final key include the length of the key and the distribution of the values of the final key. Moreover for a large scale deployment of the techniques presented, more extensive experimental evaluation is needed using representative samples of the population that would be using this system.

5. HISTORY BASED IDENTITY VERIFICATION AND MANAGEMENT

In this chapter we present *history based* identity attributes that are related to principals' past transactions that can be used by principals, together with other identity attributes, to perform identity verification and enabling SPs to make trust-based decisions concerning current transactions. One category of such systems is represented by the *reputation systems* [29, 109]. Several e-commerce SPs have built reputation systems so as to give a better idea of how trustworthy both the buyers and the sellers are. This is because the sellers are typically SPs but could also be principals in a peer to peer (P2P) environment. Sellers benefit from the use of such systems because good reputation score is likely to attract more customers. Similarly buyers may qualify for better deals and services if they have good reputation. However, most reputation systems have a major limitation in that the only information they maintain are scores and they do not typically provide information about the actual transactions a seller or buyer has made. Therefore it is important that trust be established also according to the transaction history based attributes. Information about the history can be consulted to evaluate and manage the potential risks in a given transaction.

Capturing and using transaction history for trust establishment entails addressing several challenges. First, there should be a privacy preserving methodology to guarantee ownership of the history based attributes on which the trust decisions are made. Moreover, in e-commerce applications, transaction history of individuals includes their customer profile of transactions with several SPs and such transaction history needs to be accessed by various SPs, which may use heterogenous transaction history formats. In some existing real world scenarios the SPs store transaction history in such a way that makes it impossible for other SPs to use it. Therefore the principal cannot benefit from its past transactions. Additionally, there is a lack of *user control* on his/her transaction history. The transaction history is generally stored at the SP end, and the principal may not be able to control who accesses

this information. One solution is to introduce a third party receipt management server. To this extent, we propose an extension to VeryIDX as an electronic receipt infrastructure and protocols to build and manage transaction history based attributes of principals. With such system, SPs can have access to the principal's transaction history according to the principals' permissions. The history based attributes are encoded as receipts related to the past transactions.

There are several desired properties for such a transaction history management system. First is **stealing prevention** for receipts. If a receipt R_{P_A} is issued to a principal P_A, then P_B who steals this receipt should not be able to present R_{P_A} as its own receipt. Second is the **availability** of receipts. If the transaction history is saved as cookies locally at the client machine, portability and hence the availability of such receipts is hard to achieve. VeryIDX infrastructure is based on an identity management system that makes the receipt information available to the online principals. Third is the **minimal disclosure** of the information stored in the receipts, to minimize the information revealed about the principal's transactions at the various SPs. Fourth is **user choice**; the principal should be able to select parts of a receipt based on the information needed to carry on the current transactions. Fifth is **integrity** of the principal's history based attributes. Integrity should be maintained to enable high assurance trust establishment and reputation evaluation. From the architectural perspective, a sixth desired property is that the system should be **easy to deploy** in current e-commerce systems with minimal extensions to the existing systems. The management overhead imposed on individuals should be as low as possible so to assure **usability**. The final property is that the system should support **interoperability**, in that it should be possible to use the transaction history from one SP at another SP.

We extend our approach to the use of receipts in offline in-person transactions at physical SP locations using mobile phones. In the case of using history based attributes in mobile devices the user control and minimal disclosure properties are shown to be especially important [110] and should be supported. Moreover, the computational and resource constraints of such devices should also be considered to ensure **efficient** execution of the proposed protocols.

In this chapter we explain protocols for managing transaction histories that verify the above properties.

Among our key innovations is a series of protocols for the establishment and management of principals' transaction history. These receipt protocols satisfy specific security requirements namely correctness, integrity, single submission, fairness and non-repudiation. To achieve such properties several cryptographic tools such as zero-knowledge proofs, identity-based signatures, contract signing and certified email protocols are used in the receipt protocols. All receipt protocols are privacy-preserving with respect to user consent and minimal disclosure. We provide a standard yet extensible format of e-receipts that is used in these protocols. In Appendix B describe a prototype implementation of the VeryIDX system with detailed performance analysis using such history based attributes. The architecture and design of the VeryIDX system takes into account several important considerations of a real-world e-commerce system infrastructure.

In the mobile phone context we present the main protocol that uses the cellular phone components to store and use the receipts for in-person transactions. We also analyze the mobile phone solution under several criteria including performance, portability, security and privacy.

The rest of the chapter is organized as follows. Section 5.1 provides an overview of the main approach and the key functionalities of the system with security and privacy criteria that the receipt protocols need to satisfy. Section 5.2 introduces the proposed protocols followed by a protocol analysis in Section 5.3. Section 5.4 we present the extension of the main approach in the context of offline transactions using mobile phones. In particular we present additional set of requirements specific to the use of receipts in mobile phones in Section 5.4.1, followed by the protocol satisfying the requirements in Section 5.4.2 and analysis in Section 5.4.3. In Section 5.5 we provide a summary.

5.1 Overview of the Approach

In our approach to transaction history based attribute management, the registrar (See Chapter 2 Section 2.1.1) manages principal's receipts in addition to the identity record (IdR) and provide them to SPs when needed. All receipts are stored in the principal Receipt Record (RREC for brevity) that is created for each registered principal. An e-receipt has 9 key elements, namely– TRANSACTION ID, SELLER, BUYER, ITEM, ITEM DESCRIPTION, PRICE, USER INFO, RECEIPT ASSURANCE LEVEL and TIME. The TRANSACTION ID and SELLER form a key to uniquely identify the receipt. Most of the items in the receipt correspond to those of traditional receipts except USER INFO and RECEIPT ASSURANCE. USER INFO captures only the weak attributes collected about the principal during the e-transaction. This information is used to assess the RECEIPT ASSURANCE LEVEL that the given receipt belongs to the principal claiming a given RREC. If the combination of the weak attributes uniquely identifies the principal, then the receipt assurance level of the receipt is set to 'A'. Depending on the amount of information available about the principal, the assurance level could be set to 'B' or 'U' for unknown. Receipt assurance level will be lower if conflicts are identified. For example, if the citizenship of the principal in two different receipts is different, then there is the possibility that the two different e-transactions have been executed by different principals. We ensure, by using digital signatures, that the receipts cannot be tampered with once they are issued, even by the registrar storing them.

Example 10 An example of a receipt R of principal Alice is $\langle$ 401,E-BOOK STORE, $Alice@Reg1$, BOOK, *Quantum Mechanics*,$103.27, "AMERICAN, LAFAYETTE-IN, JUGGLER", 'A', 14:34 03/12/2007$\rangle$ where 401 is the transaction ID, E-BOOK STORE is the name of the SP and $Alice@Reg1$ is the Alice's SSO ID with the registrar $Reg1$ where this e-receipt is stored. Here the RECEIPT ASSURANCE LEVEL of the USER INFO is 'A'.

To ensure minimal disclosure of the receipt attributes, in that the principal can use them as SIT attributes and create aggregate ZKPKs (See Chapter 3) of its receipt attributes, we allow the principal to extend the original receipts with Pedersen's commitments [46]. Once the commitments are enrolled at the registrar, they can then be used to create proofs

regarding properties of those attributes as elaborated in Section 5.2. The receipt extension (*x-receipt* for brevity) has the TRANSACTION ID and SELLER to uniquely identify the receipt, followed by the element tag, such as PRICE, and the corresponding cryptographic commitment. In Example 10 if Alice enrolls a commitment corresponding to the price, then she can prove that the price is greater than \$100[1] without having to reveal the exact price. We devise a logical structure called the 'wallet' that stores the principals cryptographic secrets and potentially a subset of the IdR and RREC.

Table 5.1 Summary of receipt functions.

Function	**Purpose**
Add Receipt	Once a principal has completed a transaction, it executes the 'add receipt' protocol to retrieve the receipt from the SP and store it at the registrar.
Extend Receipt	For a receipt that is already stored at the registrar the principal can create x-receipts by adding the cryptographic commitment to the original e-receipt.
Use Receipt	When the principal interact with a SP, it can use the receipts to prove properties about its past transactions. Properties required about past transactions are specified by the *trust establishment policies* of the SP.
Remove Receipt	If a receipt is unusable, expired or revoked, then the principal, registrar or SP can delete it. Once this receipt is removed from the registrar, no other copy of this receipt stored at any other can be successfully used.

Our system provides the functions listed in Table 5.1 supporting the creation, use and deletion of the receipts. The protocols implementing the functions are described in detail in Section 5.2. It is important to mention that there are specific *security* and *privacy* requirements for all these protocols. We briefly discuss such requirements in what follows.

Security requirement. Security of the receipt protocols includes five main properties.

[1]There are ZKP's that allow to prove that a committed integer satisfies an inequality, such as a given committed value x is greater than a constant A. A possible approach to accomplish this is using interval proofs [111].

1. **Correctness.** It means that if two honest parties successfully complete an e-commerce transaction, then the final receipt is constructed with the correct receipt attributes and is included in the RREC of principals' involved in the transaction.

2. **Integrity.** It refers to the tamperproofness of the constructed receipt. If any receipt attribute is modified, then it should be possible to detect the change.

3. **Single Submission.** It requires that the same receipt be not submitted more than once as two different receipts.

4. **Fairness.** It requires that the proof-of-delivery from the buyer and the proof-of-origin from the seller are available to the seller and buyer, respectively. Moreover, the protocol must be fail-safe, in that the incomplete execution of the protocol must not result in a situation in which the proof-of-delivery is available to the seller but the proof-of-origin is not available to the buyer, or vice versa.

5. **Non-repudiation.** For two-party protocols the non-repudiation property is two-fold [112]: a) non-repudiation of origin, that is, providing the buyer with irrefutable proof that the content received was the same as the one sent by the seller; b) non-repudiation of delivery, that is, providing the seller with irrevocable proof that the content of item or token received by the buyer was the same as the one sent by the seller.

Privacy requirement. The privacy requirement for the receipt protocols consists of two main properties.

1. **User Consent.** It requires that the principals be able to consent or agree to terms or conditions that may be associated with the disclosure and use of its receipt attributes. It is important that the principal has an opportunity to reject any disclosure of receipt information if required by the SP [113].

2. **Minimal Disclosure.** It requires that only the minimal piece of receipt information, as needed by the SP, is revealed.

5.2 History Based Receipt Protocols

In this section we present receipt based protocols that enable principals to enroll their receipts with registrars, and use them with SPs. More specifically, we provide detailed protocols based on two-party message exchange and cryptographic primitives such as identity based signature (IBS) and zero knowledge proof of knowledge (ZKPK). The protocols are summarized in Table 5.2.

5.2.1 Preliminary Concepts

In the following we present the notion identity based signatures that are employed in the protocols.

Identity Based Signature Scheme: We use the ID-based signature scheme derived from the Schnorr's signature scheme given in [114]. The ID-based signature scheme consists of four main protocols, namely $Setup, Extract, Sign$ and $Verify$.

The ***Setup*** algorithm consists of the follows steps. Given security parameters $k_1, k_2 \in Z^+$

Step 1: Choose a k_1-bit prime p and a k_2-bit prime q, such that $q|p-1$.

Step 2: Choose generator g of order q in Z_p.

Step 3: Choose a random $x \in Z_q^*$, and compute $y = g^x \bmod p$.

Step 4: Choose two cryptographic hash functions $H_1(\cdot)$ and $H_2(\cdot)$, such that $H_1 : \{0,1\}^* \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \rightarrow Z_q^*$.

The ***Extract*** algorithm is an interactive protocol between the principal and the Private Key Generator (PKG).

1. The principal chooses a random $r_{ID} \in Z_q^*$, and computes $R_{ID} = g^{r_{ID}}$. It sends (ID, R_{ID}) to the PKG.

Table 5.2 Summary of receipt protocols.

#	Title	Parties	Protocol Goal	Key Challenges [*Techniques used*]
1	Adding receipt (SP-principal)	P, SP, REG	Principal adds the receipt provided by a SP after an e-transaction to its RREC at the registrar.	(a) The principal's identity is verified correctly [*AgZKPK*], (b) Integrity of the receipt [*PKE*], (c) Single submission of receipt [*Session handles*]
2	Adding receipt (principal-principal)	P, REG	Both buyer and seller are principals who perform e-transaction and add their receipts to their RREC's.	(a),(b),(c), (d) Both parties should get their receipts simultaneously [*Contract Signing Protocol*], (e) Non-repudiation [*IBS*]
3	Extending receipt	P, REG	The principal creates cryptographic commitments for selected receipt attributes.	(a), (f) The extension is done correctly and on the claimed attribute [*ZKPK*]
4	Providing receipt attributes	P, SP, REG	Principal provides selected receipt attributes to SP.	(a), (g) Availability of the principals' receipts [*online Registrar*] (h) User consent on the released attributes [*Registrar portal UI*], (i) Integrity of the released attributes [*PKE*]
5	Providing receipt attribute proofs	P, SP, REG	Principal provides proof of knowledge of selected receipt attributes.	(a),(g), (j) minimal disclosure of attribute information [*ZKPK*], (k) Non-repudiation of proof [*IBS and ZKPK*]
6	Revoking a receipt	P, SP, REG	SP invalidates the principals' receipt because of the refund of the e-transaction.	(a), (l) The refund of the item and receipt revocation happens simultaneously [*Contract Signing Protocol*], (e) Receipt is removed from RREC [*Semi-trusted registrar*]

2. Upon receiving (ID, R_{ID}), the PKG does the following: 1) Chooses a random $r_{PKG} \in Z_q^*$, 2) computes $R_{PKG} = g^{r_{PKG}} \mod p$, and 3) computes

$d_{ID} = r_{PKG} + xH_1(ID||R_{ID}||R_{PKG}) \mod q$. The PKG sends (R_{PKG}, d_{ID}) to principal.

3. Principal checks $g^{d_{ID}} \stackrel{?}{=} R_{PKG} y^{H_1(ID||R_{ID}||R_{PKG})} \mod p$. If this check holds then the private key of the principal is $sk_{ID} = r_{ID} + d_{ID} \mod q$.

To ***Sign*** a message m, under the public key ID, the principal 1) chooses a random $r \in Z_q^*$, 2) computes $R = g^r \mod p$ and $\beta = H_2(ID||R_{ID}||R_{PKG}||R||m)$, and 3) set the signature to be $R_{ID}, R_{PKG}, R, \sigma$ where $\sigma = r + (r_{ID} + d_{ID})\beta \mod q$.

To ***Verify*** a signature $R_{ID}, R_{PKG}, R, \sigma$ for message m, the verifier checks
$g^{\sigma} \stackrel{?}{=} R(R_{ID} R_{PKG} y^{H_1(ID||R_{ID}||R_{PKG})})^{\beta} \mod p$

In this scheme, *non-repudiation* is achieved by step 3 of the *Extract* protocol. This is because the private key used to sign is never revealed even to the PKG involved.

Public Key Encryption: As stated in Chapter 2 we assume a public key infrastructure for the registrars and the SPs. Public key encryption (PKE) is used while encrypting the data for a particular SP or registrar, and also when data is signed by these entities.

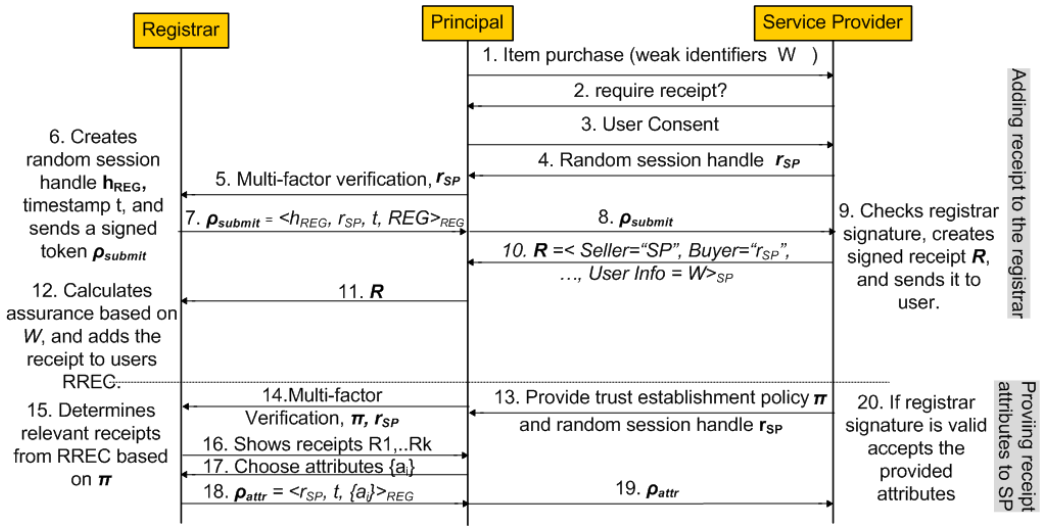


Fig. 5.1. Message flow of receipt Protocol 1 and Protocol 4.

5.2.2 Adding Receipts to the Registrar

We define two protocols for adding receipts to a registrar that vary according to the parties involved. The first protocol is applicable when a principal has conducted an e-commerce transaction with a SP and wants a receipt. The second one applies when two principals want to conduct peer to peer e-commerce transaction without an external SP.

Protocol 1: Adding Receipts generated by Principal-to-SP Transactions.
Steps 1–12 illustrated in Figure 5.1 are followed by the principal to add a receipt, generated by a SP, to its RREC at the registrar. In steps 1–4 the principal obtains a random session handle generated by the SP. In step 5 the principal conducts multi-factor identity verification as described in Chapter 2 using Aggregate ZKPK (AgZKPK) as presented in Chapter 3. Steps 6–11 in Figure 5.1 illustrate the messages exchanged among the registrar, principal and SP to retrieve the receipt. In the final step, before storing the new receipt in the RREC, the registrar calculates the receipt assurance using the procedure described below.
Receipt Assurance Assessment. To assess the assurance, the registrar verifies receipt R and compares the USER INFO (W for brevity) in the receipt, with the weak attributes (W_{user}) stored at the registrar that have a high assurance level. These weak attributes can be stored as a part of other receipts in RREC and principal information available to the registrar. Based on the overlap of this information the registrar computes the assurance that the principal who is registered is the same principal who performed the e-commerce transaction. For example if $W \bigcap W_{user} = W$ then there is a complete overlap. If W uniquely identifies the individual, the assurance level would be as high as the lower bound of the assurances of all $w_{user} \in W_{user}$. The higher the number of overlapped attributes, the higher is the assurance level. Once the assurance level is assessed the registrar adds the receipt to the principal's RREC.

Protocol 2: Adding Receipts generated by Principal-to-Principal Transactions.
Consider a case of two principals that carry out an e-commerce transaction directly with each other. A principal P_A is selling item I for price $Price$ to principal P_B. Both principals

are interested in submitting a receipt of this transaction to the registrar to extend their transaction history as a seller and a buyer respectively. This receipt would have to be constructed with the consent and verification of both P_A and P_B. We assume that principals have pre-established accounts at the registrar, in that they have a user name and password corresponding to a RREC. As P_A and P_B do not trust each other, if such a purchase/selling transaction were unsupervised then it would be difficult to settle any dispute. Therefore the following protocol is carried out to make the purchase, followed by the generation and submission of the receipt.

1. *ID-Based Signature Setup.* Principals P_A and P_B execute the IBS Scheme introduced in Section 5.2.1. Here the public IDs are the SSO IDs of P_A and P_B at the registrar. The key used to sign is only known to the principal owning that ID.

2. *Receipt and Context Agreement.* Principals P_A and P_B first sign their user ID at the registrar with their private key, using the IBS $Sign$ protocol. Once each signature is verified, by using the $Verify$ protocol, the seller and buyer names at the registrar are known. Then they need to agree on the details of the purchase involving details of price, item and other such information to construct the potential receipt. They also need to agree to provide a valid signed receipt when the transaction is complete. These terms of agreement are formalized in a *contract* C. To achieve fairness, this contract should be signed simultaneously. Therefore, a *contract signing protocol* [115] is used so that each party has a signed copy of this contract simultaneously.

3. *Purchase.* To make the purchase, principal P_B needs to provide its strong attributes such as Credit Card Number and weak attributes such as Name with Address (P_BAttr for brevity). To do this the following steps are taken [116]$-$

   (a) P_B generates a random key K and sends P_A the encrypted message $E_K(P_BAttr||C||R_{P_B})$ where C is the contract they agreed upon and R_{P_B} is a receipt signed by P_B detailing the purchase.

(b) P_A publishes a signed message requesting P_B to publish the key for E_K–encrypted message whose digest is $H(E_K(P_B Attr||C||R_{P_B}))$ by date T at location X.

(c) P_B publishes the pair $H(E_K(P_B Attr||C||R_{P_B}))$, K in X on or before date T.

The above certified email protocol prevents P_A from denying the fact that P_B provided required information for the purchase. At this point P_B's side of the purchase is made. In a similar fashion, P_A provides the resource, contract C, and a signed receipt of purchase R_{P_A} to P_B.

4. *Receipt Addition.* In this step both P_A and P_B have signed receipts. They both log onto the registrar using multi-factor verification to add their respective receipts as described in Protocol 1.

Protocol 3: Receipt Extension with Commitments.

A principal can extend receipts stored at its RREC by creating cryptographic commitments of receipt attributes. This is to allow the principals to create ZKPKs, in future transactions, based on the commitments. We omit the details of such extension as it follows directly from Protocol 1 in Chapter 3. At a successful completion of this protocol the registrar adds the signed commitment for the specified attribute to the RREC.

Protocol 4: Trust establishment with a SP using e-receipt.

Steps 13–20 in Figure 5.1 show how the principal can provide its receipt attributes to the SP to establish trust or a reputation level based on the criteria specified by that SP. This criteria may be specified as policies at the SP.

Trust establishment policies on e-receipts. The policies are specified as conditions on the receipts. We use first order logic formula (FOLF) to reason about the policies. The vocabulary Ψ_{open} contains binary predicates corresponding to receipts and receipt attributes. A complete list of these predicates is provided in Table 5.3. The SP trust establishment policy Π is a FOLF expressed in terms of Ψ_{open}.

Table 5.3 Predicates for service providers trust establishment policies.

Predicate	Arity	Arguments	Meaning
Receipt	1	receipt R	If **Receipt(**R**)** is true, then R is a valid receipt belonging to the principal who is claiming this receipt.
ReceiptKey	3	receipt R, transaction ID R.TID and seller R.SELLER	If **ReceiptKey (**R, R.T,R.S**)** is true, then R receipt can be uniquely identified using its transaction ID R.T and seller name R.S.
Seller	2	receipt R and seller information R.SELLER	If **Seller(**R,R.S**)** is true, then R has the seller name R.S.
Buyer	2	receipt R and buyer information R.BUYER	If **Buyer(**R,R.B**)** is true, then R has the buyer pseudonym R.B.
Item	2	receipt R and item tag R.ITEM	If **Item(**R,R.I**)** is true, then R has the item tag R.I.
Price	4	receipt R, price number value R.PRICE, numeric operator, number constant	If **Price(**R,R.P,o,C**)** is true, then R has the price value R.P which has a relation denoted by operator o (e.g. $=, >, <$) with numeric constant C.
Assurance	2	receipt R and assurance tag R.ASSURANCE	If **Assurance(**R,R.A**)** is true, then R has the assurance level R.A.
Time	2	receipt R and time R.TIME	If **Time(**R,R.T**)** is true, then R was issued at time R.T.

Example 11 A policy of the online-book store 'e-book' could be as follows – 'if a principal has bought a book for more than $\$80$ from 'e-book', then it is a trusted customer.' The trust establishment policy can be encoded in the logic as:

$TrustedCustomer(P) := \exists R_P(\mathbf{Receipt}(R_P) \wedge \mathbf{Buyer}(R_P, P) \wedge \mathbf{Seller}(R_P, '\,e{-}book') \wedge \mathbf{Price}(R_P, R_P.Price, >, 80))$

The SP provides the principal with the trust establishment policy Π as illustrated in step 13 of Figure 5.1, along with the random session handle that is needed to ensure freshness of the transaction. The principal then logs on to the registrar and provides this information. The registrar evaluates the policy Π to identify a list of receipts $R_1, \ldots, R_k$ that would satisfy the trust establishment criteria[2]. Once the receipts are identified the registrar provides a way for the principal to select the attributes $[a_{RREC_i}]_{R_t}$ from receipt R_t, where $1 \leq t \leq k$, $1 \leq i \leq n$, and n is the total number of attributes needed to satisfy Π. The principal is also given an option to add more receipts from its RREC if it desires to do so. The principal also provides the random handle r_{SP} to the registrar. Given this information, the registrar constructs the signed attribute token $\rho_{attr} = \langle \{[a_{RREC_i}]_{R_t}\}, r_{SP}, t \rangle_{REG}$ where t is the current timestamp. The registrar sends ρ_{attr} to the principal. Finally the principal provides ρ_{attr} to the SP. The SP verifies the attributes and provides the service accordingly.

Protocol 5: Trust establishment with a SP using x-receipt.
If a principal does not want to provide clear attributes from the receipts and instead wants to prove properties of the receipt attributes, it can use the enrolled cryptographic commitments of the x-receipts to create proofs of such properties. The policies for this kind of trust establishment can be expressed as follows.

Trust establishment policies on x-receipts. For the cases in which the trust establishment criteria are related to *cryptographic proofs* of receipt attributes belonging to the principal, the SP uses an extension of policy vocabulary Ψ_{open} denoted as Ψ_{proof}. Ψ_{proof} also has binary predicates but unlike Ψ_{open} the attributes specified do not have to be revealed in clear. Instead ZKPK of those receipt attributes need to be provided by the principal. For each of the predicates listed in Table 5.3, there is an equivalent predicate for the Ψ_{proof} vocabulary, pre-pended by the letter 'x'. For example **xSeller, xItem,** and **xPrice**. If in Example 11 the clear attributes are not required, instead the cryptographic proofs of those attributes are sufficient, and then the same trust establishment policy can be written as:
$TrustedCustomer(P) := \exists R_{xP}(\mathbf{xReceipt}(R_{xP}) \wedge \mathbf{xBuyer}(R_{xP}, P) \wedge \mathbf{xSeller}(R_{xP}, ' e-$

[2]The evaluation is possible only if the RREC has the value of the attributes needed to satisfy Π in clear.

$book') \wedge \mathbf{xPrice}(R_{xP}, R_{xP}.Price, >, 80))$

Using such policies, the precise steps of the protocol are described as follows.

1. *SP Policy.* The SP provides the trust establishment policy requiring ZKPK Π_x together with a random handle r_{SP} and sends it to the principal.

2. *Principal retrieves receipt commitments.* It is required that the principal has created a commitment for each of the receipt attributes for which it has to construct a ZKPK. Assuming that these commitments are created for each such attribute using Protocol 3, the next step is to retrieve these commitments.

   The principal logs on to the registrar to access its RREC. The registrar then evaluates the policy Π_x to identify a list of receipts that would satisfy the trust establishment criteria based on the attribute information available in clear. Once the receipts are identified the registrar provides a user-interface for the principal to add additional receipts if needed. Let the resulting list of selected receipts be $R_1, \ldots, R_k$. The principal then selects the attributes $[a_{RREC_i}]_{R_t}$ with the corresponding commitments $[C_{RREC_i}]_{R_t}$ from receipt R_t where $1 \leq t \leq k$, $1 \leq i \leq w$ and w is the number of commitments needed. We simplify the notation of the commitment and represent it as $C_1, \ldots, C_w$. The principal is also given an option to add more receipts from its RREC if the principal desires to do so. The principal also provides the random handle r_{SP} to the registrar.

   Given this information the registrar constructs the signed commitment token $\rho_{commit} = \langle\{([C_{RREC_i}]_{R_t})\}, r_{SP}, t\rangle_{REG}$ where t is the current time stamp. The registrar sends ρ_{commit} to the principal.

3. *Proof submission of principal's x-receipt attributes.* The principal performs AgZKPK with the SP to provide proof of knowledge of the required attributes. Only the principal knows the random secrets and the actual attribute values corresponding to each of the committed receipt attributes. The proof consists of the following two key steps.

   (a) *Principals' aggregation*. Consider that the SP has challenged the principal to prove knowledge of commitments $\{C_i\}$ where $1 \leq i \leq w$. The principal computes

$C = \prod_{i=1}^{w} C_i = g^{a_1+\cdots+a_t} h^{r_1+\cdots+r_t}$, where a_i and r_i are the attribute and secret random corresponding to the commitment c_i respectively. The principal sends C, ρ_{commit} to the verifier.

(b) *Zero-knowledge proof of aggregate commitment*. The principal and the registrar carry out the following AgZKPK protocol:

$$PK\left\{(\alpha,\beta)\colon C = g_1^{\alpha} h_1^{\beta},\ \alpha,\beta \in \mathbb{Z}_q\right\}$$

where $\alpha = \{a_1 + \cdots + a_t\}$ and $\beta = \{r_1 + \cdots + r_t\}$. If the ρ_{commit} is constructed correctly to satisfy Π_x and AgZKPK in step 3 is successful, then the principal proof is considered correct and the trust is established.

Protocol 6: Revoking a receipt

By revocation of a receipt we mean the removal of the receipt from the principals' RREC. We consider three cases for the revocation of a receipt depending on the party revoking the receipt, namely the principal, the registrar and the SP.

The first two cases are trivial. For the principal case, the principal is required to log onto its account using multi-factor identity verification to access its RREC. Once logged in, our system provides a way for the principal to remove any of the receipts from its RREC. For the registrar case, the registrar may define the criteria for removing receipts. For example, the registrar may revoke receipts that are more than 100 days old. The registrar periodically checks the RREC to see if the receipts are compliant to its criteria to retain the receipts. If not, the registrar asks the principal to remove the specific receipts within a given time period, after which the receipt is removed by the registrar itself.

For the SP revocation case we consider an interesting scenario where the principal needs to return the purchased item from a SP and the SP may provide a refund. More importantly this SP needs to ensure that the receipt stored from the previous transaction gets void and the principal needs to ensure that it gets the refund. The following steps are needed to do this revocation.

1. *Principal retrieves receipt from its registrar.* The principal logs on to access its RREC. It identifies the receipt R which is to be revoked once it returns the item relevant to the purchase identified in the receipt. The registrar then constructs the signed token $\rho_{revoke} = \langle R, P, t \rangle_{REG}$. It sends ρ_{revoke} to the principal where P is the SSO ID of the principal, and t is a timestamp to ensure freshness.

2. *Principal requests revocation from SP.* The principal then signs the $\langle \rho_{revoke} \rangle_P$ using IBS $Sign$ protocol and provides this token to the SP. The SP verifies the signature using the public key of P and the IBS $Verify$ protocol. Only if the verification is successful, the revocation protocol proceeds.

3. *Principals' refund and SPs revocation.* The principal and SP agree on a contract C using a simultaneous contract signing protocol, which would state that for the transaction identified by the receipt R in ρ_{revoke} the principal will provide the SP the identifier i of the purchased item, and the SP will provide the refund f applicable to that purchase. The item identifier should not be a sensitive value, but instead a public service number for the item purchased. For example i could be a pin. Once the pin is revoked, no other principal can use it to access the same resource. Once this contract is agreed upon, the three steps as in Protocol 5 Step 3 are executed with message $E_K(i||C)$ where C is the refund contract they agreed upon. Using these steps the principal can request revocation and the SP cannot deny the principal did revoke its purchase. Then the SP sends the principal a token $\alpha = [f, \rho_{revoke}]_{REG}$ encrypted with the registrar's public key. The principal sends α to registrar. The registrar removes the receipt identified in ρ_{revoke} and sends the refund f to the principal.

5.3 Analysis

In this section we present an analysis of the receipt protocols based on the security and privacy requirements introduced in Section 5.1. For ease of understanding, a summary of the cryptographic techniques used in the various protocols that provide the various security and privacy properties are given in Table 5.4.

Table 5.4 Analysis of the security and privacy requirements of the receipt protocols based on cryptographic building blocks.

#	SECURITY					PRIVACY	
	Correctness	**Integrity**	**Single Submit**	**Fairness**	**Non-repudiation**	**User Consent**	**Min. Disclosure**
2	IBS, AgZKPK, Contract Signing	IBS	Session handles	Contract Signing	IBS, Certified Email Protocol	IBS, Contract Signing	AgZKPK, SSO ID
3	ZKPK	IBS	ZKPK	N/A	IBS	Commitment, ZKPK, IBS	N/A
5	AgZKPK	PKE	Commitments	N/A	ZKPK transcript	ZKPK	ZKPK
6	PKE	PKE	PKE	Contract Signing	IBS, Certified Email Protocol	IBS, Certified Email	SSO ID

Property 1 *(Security of Receipt protocols) All receipt protocols satisfy the security criteria namely 1) Correctness, 2) Integrity, 3) Single Submission, 4) Fairness and 5) Non-repudiation properties*

For all the protocols, the multi-factor identity verification at the registrar using AgZKPK prevents identity theft attacks as described in Chapter 3. As such, if an adversary is able to impersonate a given principal P and authenticate using k random commitments, then that would imply that this adversary was able to steal the corresponding $2k$ secrets of P to construct a valid proof. Such compromise can occur with a low probability and hence the login at the registrar is reliable. In addition the evaluation of the RECEIPT ASSURANCE LEVEL based on the principals' weak identifiers stored at the RREC's, also mitigates the risk of the adding and using incorrect receipts. For each of the detailed protocols, the security criteria are discussed below.

In Protocol 2, *correctness* of the buyer and seller information is achieved by multi-factor identity verification and use of IBS. The IBS scheme is provably secure based on the Schnorr's signature scheme [117] in a random oracle model. If the signatures are correct, it would imply that the buyer and seller information provided for the receipt is correct. For the correctness of the receipt attributes, the contract signing protocol [115] is used. The principals agree on a set of attributes relevant to the e-transaction and use it to carry out the protocol. *Integrity* of the e-receipts is achieved by IBS signatures [114] on the final receipts R_{U_A} and R_{U_B} provided by each principal. The *single submission* is ensured based on the session handles included in ρ_{submit} token used during the final addition of the receipt at the registrar. The *fairness* is proven and achieved because of the use of the simultaneous contract signing protocol [115]. Finally *non-repudiation* is achieved because of the use of the IBS. This is because the IBS scheme used achieves the Girault's trusted level 3 [118] that implies the that private key generator (i.e. the registrar) does not know, or cannot easily compute, the principals' private keys. Moreover, the certified email protocol given at step 3 requires that the requests and keys shared in step 3, be published, therefore the parties cannot deny carrying out the transaction.

In Protocol 3, *correctness* is ensured because of the use of the ZKPK while creating the commitment. The *integrity* and *non-repudiation* properties are achieved because of the use of the IBS signature and AgZKPK at the time of identity verification.

In Protocol 5, *correctness* is achieved using mutli-factor identity verification, followed by the AgZKPK on the commitments identified in token ρ_{commit}. For the *integrity* of ρ_{commit}, the public key signature of the registrar on this token is used. The tamperproofness of ρ_{commit} prevents adding any additional commitment of an attribute which may not belong to a valid receipt. Thus the *single submission* of the attribute commitment is achieved.

In Protocol 6, ρ_{revoke} is first signed by the registrar using PKE, and eventually by the principal using IBS. These signatures ensure *integrity* of the receipt that needs to be revoked. The signed token $\langle \rho_{revoke} \rangle_U$ and the timestamp t prevent receipt from being re-submitted by an adversary. Using ρ_{revoke} also helps in the *single submission* of receipt

revocation requests. Finally the *fairness* and *non-repudiation* properties are achieved as in Protocol 2.

Property 2 *(Privacy of receipt protocols) All receipt protocols preserve the privacy criteria, namely 1) user consent and 2) minimal disclosure of principal receipt attributes and other principals identity information, as described in Section 5.1.*

In Protocol 2 the *user consent* is captured using the IBS signatures, and the contract signing protocol. This is because only the principal is assumed to have the secret key for executing the $Sign$ protocol. Moreover the terms and conditions of the e-transaction are encoded in the contract that is signed by each participant principal. The protocol also ensures *minimal disclosure* which is achieved by the use of random session handles. Even if both principals' identity is verified with mutli-factor identity verification using AgZKPK at the registrar, the principals do not learn any other information besides the SSO ID of each other, and the information required for the e-transaction to occur.

In Protocol 3 the *user consent* is ensured when the principal creates the cryptographic commitment followed by the IBS signature and the ZKPK on the committed value. Subsequently in Protocol 5, *user consent* is captured based on the ZKPK which can only be performed if the principal provides its secrets associated with the receipt attributes on which the proofs are formed. The ZKPK also helps in the *minimal disclosure* because of the security of Pedersen commitment [46] that relies on the hardness of the discrete log problem. Finally in Protocol 6, the IBS and the certified email protocol ensures that the principal participates in the revocation procedure and that there is *user consent*. During this protocol no other information other than the principals' SSO ID is revealed to the SP conducting the revocation, thus ensuring *minimal disclosure* of principals attributes.

5.4 Receipts in Mobile Phones

In this section we extend the above approach to show how such receipts can be used in physical in-person commercial transactions. It is desired to have a portable device which can store and compute ZKP's in addition to communicating the the physical SPs. We

use Near Field Communication (NFC) enabled cellular phone devices to store and do the necessary computations to execute the receipt protocols. NFC is a standards-based, short-range ($\sim$15 centimeters) wireless connectivity technology that enables two-way interactions among electronic devices, allowing users to perform contactless transactions, access digital content and connect electronic devices [30]. First we present an additional set of requirements on receipt usage in the mobile device context, followed by protocols satisfying those requirements. We also provide additional analysis based on the use of receipts in mobile phones.

5.4.1 Additional Requirements

In this section we highlight specific requirements related to user control that are important in the mobile device based in-person transactions [119]. In our context the user control on his/her receipts stored on the mobile device is ensured by satisfying the following specific requirements.

1. **Condition-based Receipt Retrieval:** Retrieval of receipts from the mobile device should not be unconstrained, rather it should be driven by conditions defined by user preferences or SP policies. These conditions should be taken into account while making queries on the RREC or subset of RREC stored on the mobile device.

2. **User Consent:** The individual should provide explicit consent or be aware of the data being revealed from the mobile device.

3. **Minimal Disclosure:** Similar to the privacy requirement for online transactions given in Section 5.1, in the context of mobile phones, the individual should be able to disclose to the SP the minimal information about the receipt attributes that is needed as per the SP service policies.

Satisfying the above requirements in addition to the security and privacy requirements provided in Section 5.1 is non-trivial, because of technical and practical challenges. Our

overall goal in this section is, by satisfying the requirements, to support flexible and portable receipt based transactions, as in the following example.

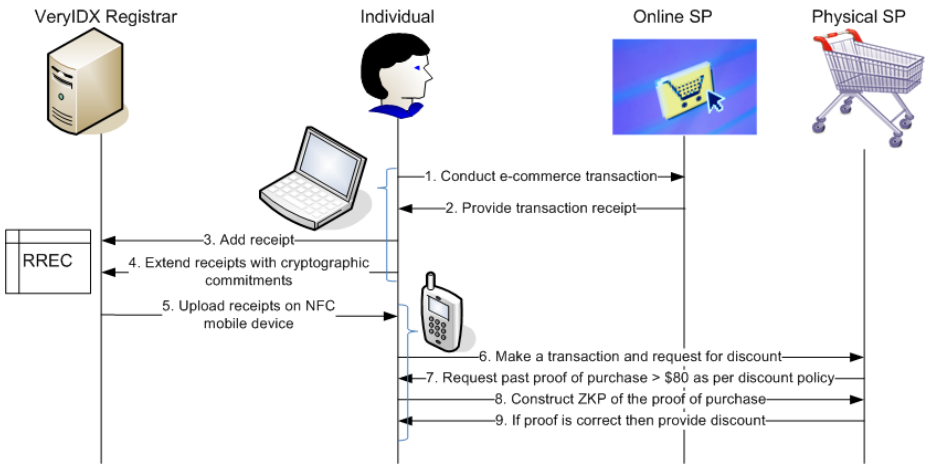


Fig. 5.2. Example scenario of NFC mobile phone based receipt management.

Example 12 Assume an individual Alice, denoted by her SSO ID $Alice@Reg1$ at a registrar Reg1, conducts an e-commerce transaction with SP $eFollets$ to buy a book for the price of $134.65. The receipt of this transaction is uploaded to the RREC in steps 2 and 3 according to the receipt protocols detailed in Protocol 1. Using Protocol 3 Alice can extend the receipts to establish cryptographic commitments corresponding the one or more receipt attributes. Alice extends her set of receipt attributes to create a commitment (step 4) on the price of the receipt received in step 2. For portable usage of the receipts in the RREC, a subset $\mathcal{R}$ of the receipts in RREC is uploaded to the NFC mobile phone by Alice in step 5.

Alice then decides to use her receipts at a physical SP shop *Follets* to qualify for a particular discount that requires her to have performed a (possibly electronic) commerce transaction involving buying an item from the $Follets$ or $eFollets$ for more than 80$. The device has the capability to retrieve the appropriate receipts based on the conditions specified by the SP. The receipt information is not passed from the NFC mobile phone to

the SP without explicit user consent of the user. Moreover to ensure minimal disclosure, Alice can prove using ZKPK that the receipt from the $eFollets$ transaction was greater than 80$ without showing this value in clear. This is depicted in step 8. If the proof is correct then Alice qualifies for the discount offered by $Follets$.

In the above example, $Follets$ may also wish to ensure that the receipts are actually owned by the individual presenting the receipts. Using the information stored in the mobile phone, which is signed by the registrar, $Alice$ can prove to $Follets$ that she owns the RREC that contains the receipt attributes being presented. The correctness and integrity of the receipt attributes involved is ensured by the registrars' signature on the receipt attributes and multi-factor identity verification using AgZKPK. In this manner the user control and security requirements of the use of receipts in mobile phones is ensured.

5.4.2 Receipt Protocol for Mobile Devices

Table 5.5 Nokia NFC mobile phone components.

#	Component	Symbol	Description	Usage in Receipt Protocol
1.	MIDlet	Ph^{mid}	Detailed description in § 5.4.2	Main applications running the receipt usage protocol
2.	Phone Memory	Ph^{mem}	11 MB memory	To store secrets, IdR and RREC tuples
3.	External Memory	Ph^{xmem}	2GB memory	To store secrets, IdR and RREC tuples
3.	Smart Card	NFC_{sc}^{dev}	72KB memory	To store secrets
4.	Mirfare Tag	NFC_{tag}^{dev}	4KB memory	Used for communication with the SP
5.	Modem/ Antennae	-	Communication components	Used for communication with the SP

In this section we provide the protocol that satisfies the requirements highlighted in the previous subsection. As we will see, the protocol description refers to the Nokia NFC

Table 5.6 Summary of mobile device based receipt usage functions.

Requirement	**Function**	**Description**
Condition-based Receipt Retrieval	QueryRREC(RREC, $conditions$)	This function returns the receipts in the RREC which satisfy the conditions listed in $conditions$ which specified by user preferences and/or the SP service policy.
User consent	UserInterface (Receipts)	This function is responsible for the user interaction interface involved in the selection and submission of receipt attributes.
Minimal Disclosure	CreateProof($ReceiptIDs$, $Commitments$, $Tags$)	To create AgZKPK on the receipt attributes indicated by $ReceiptIDs$ and $Tags$, along with the list of associated commitments $Commitments$ stored.
Ownership	VerifyID(Verification Policy)	The function returns true only if based on multi-factor identity verification using AgZKPK on the information present in the RREC and IdR is successful.
Receipt Usage	VerifyReceipts(Receipt Policy) (§ 5.4.2)	The protocol describes how the receipts can be used to satisfy the receipt based on the SPs trust establishment policy.

cellular phone architecture. We thus begin our presentation with a brief overview of the key components of the mobile phone that are utilized in the protocols and some basic functions implemented on the phone itself for the receipts usage. Following that we provide the protocol for privacy preserving usage of receipt attributes.

Preliminary Concepts regarding Cell Phone Architecture

We use a Nokia 6131 NFC cell phone (Ph^{NFC}) [30] to store and use portable receipts for in-person transactions. We assume that the SPs have a NFC reader (denoted by NFC^{SP}_{reader}) that transmits and receives messages from a NFC cell phone. The phone is integrated with a NFC device and thus contains both reader and writer to receive and send

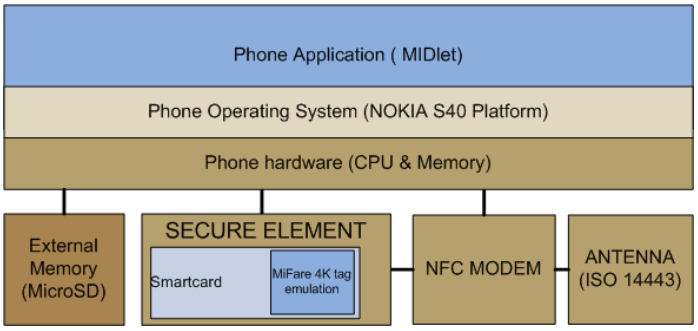


Fig. 5.3. NFC mobile phone components.

messages from/to a SP. The components of the Ph^{NFC} are shown in Figure 5.3 and briefly described in Table 5.5.

The main component used to manage and use receipts is given by the MIDlet suite. The MIDlet suite consists of a Java Application Descriptor (JAD) and MIDlet. The JAD describes the MIDlet applications in the suite. A MIDlet (denoted by Ph^{mid}) is a Java program that runs on the Mobile device. A Ph^{mid} can be installed onto a phone and use Mobile Information Device Profile (MIDP) in Java 2 Platform Micro Edition (J2ME) [120]. The MIDlet operates in a sandbox [121] that restricts the available APIs to a limited set. Once the Ph^{mid} has been deployed on the cell phone, it uses the cell phone's CPU and memory. To access the secure elements of 6131 NFC (NFC_{tag}^{dev} and NFC_{sc}^{dev}) Nokia requires that only signed MIDlet's can access the secure elements. Such signed MIDlet's are called *trusted MIDlet's*. Security for trusted MIDlet suites is based on *protection domains* [120]. Each protection domain defines the *permissions* that may be granted to a MIDlet suite in that domain. These permissions are checked by the implementation prior to the invocation of any protected function. In Section 5.4.3 we provide additional details on the protection domain and how it applies to the protocol presented next.

The functions supported by Ph^{mid} are briefly described in Table 5.6. Some of the functions can be implemented using existing technologies. For example, the QueryRREC

function can be implemented using tools for querying the mobile database [122]. Also, several techniques have been proposed [123] for ensuring a usable UserInterface for mobile phone applications. The CreateProof and VerifySig functions rely on the Aggregate ZKPK and signatures introduced Chapter 3. The VerifyID and VerifyReceipts functions are novel and are based on the articulated cryptographic-based solution. Both functions enable verification of attributes, and are thus similar. In the former, the identity attributes in the IdR are proven to the verifier and in the latter the receipt attributes in RREC are used to prove history-based attributes. In this section, we focus on how the VerifyReceipts function is implemented on the Ph^{NFC}.

Managing Receipts in the NFC Cell Phone Device

Now we present the protocol for providing receipts using Ph^{NFC}. We focus our attention on the key steps specific to the NFC device itself, and omit the details on the cryptographic protocols involved for the correctness and integrity of the receipts[3]. To show where such cryptographic protocols are used, we make the appropriate calls to functions as listed in Table 5.6.

Adding receipts to the NFC device is straightforward as the individual can select the digital receipts from his/her RREC which, in turn, is saved in the external memory of the NFC device. More specifically, each time an individual obtains a receipt in a physical SP location, then this receipt can use standard digital data communication technology such as bluetooth, Infra Red communication(IR) and USB cable [30] to upload this receipt. On the contrary, the verification protocol has several interesting challenges. Recall that the protocol is carried out by the individual to provide the proofs of receipt attributes required to satisfy the SPs trust establishment policies π_{SP} as illustrated in Protocols 4 and 5 in Section 5.2. SPs specify policies that describe the conditions that need to be satisfied by a given receipt before the individual qualifies for a particular request (example a discount). Such conditions will be encoded into queries on the receipt records stored in the individu-

[3]See Chapter 3 for details on the cryptographic protocols.

als' mobile device. The main steps of the VerifyReceipt protocol to engage for satisfying queries of this kind are provided in Protocol 7 that is explained in the following.

As a pre-requisite, individuals' Ph^{NFC} is initialized with a set of receipts $\mathcal{R}$ uploaded before carrying on the protocol. In the first step the individuals' cell phone tag denoted by NFC_{tag}^{dev} captures π_{SP} sent by the SPs transceiver NFC_{reader}^{SP} [4]. The NFC_{tag}^{dev} transfers this policy to the cell phones main memory Ph^{mem} in step 2. Subsequently, in step 3 NFC_{tag}^{dev} triggers an event to the cell phones computational unit to initiate the Ph^{mid} MIDlet to run the receipt queries. Ph^{mid} calls the function QueryRREC to evaluate the potential receipts in $\mathcal{R}$ that can satisfy the conditions in π_{SP}. As a result the eligible receipts $\mathcal{R}' \subset \mathcal{R}$ is retrieved from the Ph^{NFC} and displayed on the cell phone's screen. In step 5, the Ph^{mid} calls the user interface related function UserInterface that allows the user to choose the receipt attributes from $\mathcal{R}'$ that the user wishes to show in clear (L_1) or create a ZKPK (L_2).

In the next step, the main Ph^{mid} initiates a new MIDlet called Ph^{midc} that runs in a protected domain with restricted permissions. This is because Ph^{midc} uses cryptographic secrets associated with the receipt attributes to create receipt proofs. The receipt proofs are created in an aggregated manner using the function CreateProof in step 8. This results in the aggregated proof called $AgProof$. Ph^{midc} sends the $AgProof$ to Ph^{mid}. The receipt attributes and proof are concatenated in step 9 to obtain the final token $\mathcal{F} := L_1||L_2||AgProof$ where L_1 is a list of receipt id's, attribute values signed with the registrar's key, and the corresponding tags it wants to reveal in clear; and L_2 is a list of receipt id's, commitment values signed with the registrar's key, and of the corresponding tags the individual wants to prove ownership. Using the UserInterface function the individual approves sending this information to the SP. On receiving user consent, $\mathcal{F}$ is sent via the NFC_{tag}^{dev} to be read by the NFC_{reader}^{SP}. If the receipt attributes and proofs provided satisfy the conditions defined in the SPs policy (π_{SP}), then the individual receives the services as specified in π_{SP}.

[4]NFC Transceiver is a device that can transmit as well as receive data using NFC.

Protocol 7 [VerifyReceipt] User providing receipt attributes from Ph^{NFC} to SP

Require: SP trust establishment policies π_{SP}, user receipts $\mathcal{R}$ on Ph^{NFC}.
Ensure: The user's Ph^{NFC} and the SPs $NFC^{SP}_{reader}$are located at a close proximity.

1: $NFC^{SP}_{reader} \xrightarrow{M1} NFC^{dev}_{tag}$ $[M1 = \pi_{SP}]$
2: $NFC^{dev}_{tag} \xrightarrow{M2} Ph^{mem}$ $[M2 = \pi_{SP}]$
3: $NFC^{dev}_{tag} \xrightarrow{M4} Ph^{CPU}[M4 =$initiate Ph^{mid} event]
4: Ph^{mid}[uncritical domain] executes QueryRREC$(\mathcal{R}, \pi_{SP}) \leftarrow \mathcal{R}'$
5: Ph^{mid}[uncritical domain] executes UserInterface ($\mathcal{R}'$)
6: { User chooses $L_1 := \{R_i, a_k, attr_k\}$ which is a list of receipt id's, signed attribute values with registrars key, and the corresponding attributes it wants to reveal in clear
7: User chooses $L_2 := \{R_i, C_l, attr_k\}$ which is a list of receipt id's, signed commitment values with registrars key,and the corresponding attributes it wants to prove }
8: Ph^{midc}[critical domain] executes CreateProof (L_2) $\leftarrow AgProof$
9: Ph^{mid}[uncritical domain] executes UserInterface to provide consent to final token $\mathcal{F}$:= $L_1||L_2||AgProof$
10: $Ph^{mid} \xrightarrow{\mathcal{F}} NFC^{dev}_{tag}$
11: $NFC^{dev}_{tag} \xrightarrow{\mathcal{F}} NFC^{SP}_{reader}$

5.4.3 Analysis of Receipt Protocol for Mobile Devices

In this section we analyze the receipt usage using the Ph^{NFC} with respect to performance, portability, security and privacy criteria. We focus on the applications running on the phone executing the receipt usage protocol and discuss how we use specific capabilities of the phone to achieve the desired properties.

Performance

One of the key features required of a MIDlet is that it should run efficiently on the mobile phone platform. One main factor that would impede the performance is the use of large numbers to perform the ZKPK computations. Because creating ZKPK proof computation done at the MIDlet uses computations on large integers ($\sim$128 Bytes), it may be expected that the time taken to compute the ZKPK to be proportional to the number of receipt attributes involved. However, using AgZKPKs it takes almost constant time for ZKPK generation even as the number of attributes being proven increase. This is because

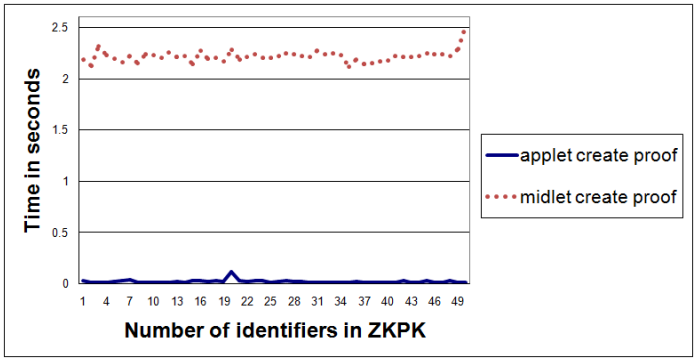


Fig. 5.4. Comparison of proof create in Midlet versus Applet.

AgZKPK has a constant number of exponentiations while providing proof of knowledge (See Chapter 3 Table 3.2). This claim is confirmed by the following experimental test results.

The performance testing is based on the Nokia 6131 NFC mobile phone [30]. The Ph^{midc} has the size 17 KB. A graph displaying the amount of time it takes for the aggregated proof for number of identifiers ranging from [1,. . .,50] is provided in Figure 5.4. Overall, the estimated time for creating a proof varying with one to fifty attributes is 2.22 seconds on an average. We compared the amount of time it takes to create the proof in the Ph^{mid} versus the time it takes to create the proofs in an online transaction using a JAVA Applet [124] which is on an average .020 seconds. The increased number of identifiers being proven does not increase the time. We also compared the time the SP takes to verify these proofs at the server (which is an Intel Pentium D CPU 3.0 GHZ and 1G RAM and runs the Windows XP Operating system) to the time it takes to create it using an Applet in Figure 5.5. On an average the SP takes 0.103 seconds to verify aggregate proof of 50 identifiers.

We also analyzed the communication costs by measuring the number of bytes sent to the verifier when the user provides a proof of identity attributes. The amount of time taken

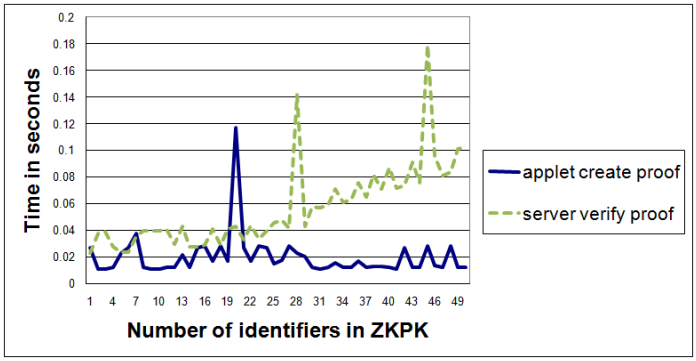


Fig. 5.5. Comparison of proof create versus proof verify.

to transfer the messages relies on the network speed in case of online transaction. Figure 5.6 illustrates the increase in message size with increased number of identity attributes being proven. The size of the aggregate proof is approximately the same ($\sim$167 Bytes) but the other information associated with the proof such as the commitments, and the tag information (See step 9 of Protocol 7) increases with the number of identifiers. The message size for each round with a given number of identity attributes was averaged over 3 runs. The number of bytes increase 161 Bytes on an average as the proof includes one more identifier. We use the `tcpdump` tool to get the the size of TCP data that is transmitted. For one identity attribute $\sim$1582 Bytes is sent on an average.

Portability and Interoperability

Portability allows users to have multiple devices (such as mobile phones and external storage devices) implementing the protocols, thus enabling user choice not only on the attributes but the device itself. Portability is achieved through adherence to standards and use of MIDlet's for applications. MIDlets can be copied to other platforms and used to manage receipt and other identity attributes in the RREC and IdR respectively.

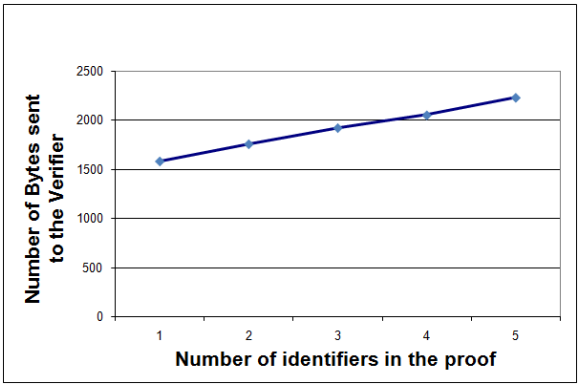


Fig. 5.6. Message size analysis with increased number of identifiers.

The receipts can be stored and added as per the memory capability of the Ph^{NFC} and the Ph^{xmem}. Our implementation of the wallet in the Nokia 6131 NFC model can use up to 11MB of Ph^{mem} and up to 2GB of Ph^{xmem}. The unit size of a IdR or RREC receipt tuple containing one commitment is 4KB. The Ph^{xmem} can hold up to 50,000 receipts for 2GB memory and the maximum number of receipts that can be stored into phone's internal memory is 2750, although this number will vary based on the memory taken by other files in the phone, such as the multimedia files. Individuals need to upload only a subset of records in RREC and IdR. The wallet contains the secrets needed for the proof generation. With any modification to the secrets, the proof verification will fail so the integrity needs to be ensured. To ensure confidentiality of the secret residing in plain text in the Ph^{xmem} the user can lock the memory card with a password. It is however not required that the password is re-entered every time an access call is made to the wallet file in the Ph^{xmem}. Further mechanisms to ensure confidentiality is a part of our future work.

Regarding interoperability, the stored secrets in the user's wallet can be transferred to various devices such as Ph^{mem} Ph^{xmem} or even other Ph^{NFC}. The NFC phone supports InfraRed, bluetooth and USB cable connection for this data transfer. Ph^{xmem} can be used from another cellphone as the user decides to use a different mobile device. Keeping wallet

in Ph^{xmem} helps our system to maintain interoperability, in that the memory card can be used from some other cell phone using the Ph^{xmem} [30].

Security and Privacy

Fundamental security and privacy properties such as minimal information disclosure and non-replay of proof of ownership are inherited by the employed AgZKPK protocols. In the following we instead focus on the possible attacks on the MIDlets and on how our proposed protocols ensure user consent when releasing receipts attributes.

The integrity of the MIDlet is ensured by using the trusted MIDlet suite for all the applications running our protocols. The trusted MIDlet suite is composed of signed code so as to ensure the integrity of the applications running on the Ph^{NFC}. An important aspect of MIDlet is that they run in a sandboxed environment [121], providing the necessary isolation of the memory usage between $Ph^{mid}$and Ph^{midc}.

One possible attack on the MIDlets can occur if an attacker intercepts the MIDlet application during proof creation, to either read the cryptographic secrets to compromise the confidentiality of such secrets, or write to the MIDlet during proof creation resulting in possibly incorrect proofs. To mitigate this attack, we run the information critical MIDlets (Ph^{midc}) in a restricted environment with no connectivity with external devices so the attacker cannot use the excessive permissions or open ports, to access the Ph^{midc} to exploit any potential vulnerabilities. More specifically we consider two types of *trusted domains*. If the MIDlet needs to access cryptographic secrets, such as Ph^{midc} in Protocol 7, then it is run in a restricted domain called the *critical protected domain*. This domain helps protect against the interception of possibly malicious programs to retrieve the secrets used in a given computation [30]. If other functionalities are needed, for example Ph^{mid} in Protocol 7 needs connectivity with the NFC^{SP}_{reader}, then it runs in the *uncritical domain* that contains a set of permissions to access the Ph^{NFC}'s resources. The set of permissions are often called together as function groups. An example of a function group assigned to

Ph^{mid} is *Local Connectivity* that contains permissions related to connection via local ports such as NFC to do the necessary communication with NFC^{SP}_{reader}.

Another attack, related to user's privacy is the potential opportunity for a malicious SP to access the receipt and other identity attributes from the user's mobile phone without explicit consent of the individual. To prevent this threat it is crucial to ensure user control [110]. An individual should provide explicit consent to every transaction or attribute receipt exchange. The user consent is attained in steps 5 and 9 of Protocol 7. Internally, in the Ph^{NFC}, the UserInterface function of Ph^{mid} displays the potential receipt attributes the individual can use in a given transaction. Based on the individuals' choice, the list of attributes L_1 and L_2 are constructed. This is followed by cryptographic operations computed by the Ph^{midc} whose permissions are set requiring user input (denoted by *User Permission* in [120]) to execute the CreateProof function. Finally, in step 9, before the receipt attributes and proofs are revealed to the SP, this information is checked using the UserInterface function of the Ph^{mid}. In this manner, the user consent property is achieved and the individual maintains a level of control over which identity attributes are released to a given SP.

5.5 Summary

In this chapter we have presented the concept of history based identity attributes encoded as receipts related to online transaction histories of individuals and protocols to build and manage such attributes. We show how the protocols can ensure several desired security and privacy properties and be used along with other identity attributes for multi-factor identity verification, during the usage of such history based attributes. Given our approach individuals' online activity can be used to generate reliable identity information that can be managed and used as any other identity attributes to evaluate trust relationship based related properties such as reputation. We also show how the receipts can be portable, and used with mobile phone devices. In the mobile identity context we further analyze and show that are protocols are effective to achieve the desired performance, security and pri-

vacy properties in the system. In essence, the receipt protocols presented in this chapter provide a flexible and privacy preserving methodology to use history-based attributes in the VeryIDX framework.

There are specific assumptions to consider while employing the history based protocols. First, is the participation of the SPs in issuing receipts to individuals as specified in the protocols. Second, at the time of verification it is assumed that the SPs define policies based on such receipt attributes and attribute properties which can potentially be proven in zero knowledge.

The receipt protocols allow *user choice* when receipts are revealed to a given SP. Therefore if a user does *not* provide a receipt then it does not imply that the user did not execute a particular transaction. More specifically, the SPs do not gain knowledge about all potential transactions executed by the user, but instead those that the user chooses to reveal. Trust establishment based on the knowledge of all possible transactions of a given user will require additional mechanisms such as profiling.

An important aspect for successful deployment of protocols related to e-commerce is to analyze the constraints and requirements of the various e-commerce applications. For example, the secure electronic transaction (SET) [125] protocols that provided mechanisms to allow SPs to substitute a certificate for a user's credit-card number, failed to be implemented because of several practical considerations. A first consideration was with respect to the cost and complexity for SPs to support such protocols, especially given the presence of simpler alternatives such as SSL [126]. In addition it was cumbersome to install client software and allow client-side certificate distribution. In our approach we provide a flexible mechanism to allow various types of transactions as per the capability and requirements of the system. Moreover we show that there is minimal computational overhead and need for client software in our prototype implementations. However, to be practical, additional studies of human computer interaction [127], market acceptance and other business requirements are needed.

6. RELATED WORK

In this chapter, we survey work related to our thesis. In our thesis we present an infrastructure and several techniques for the protection of digital identity in IdM systems. The main innovative features we have proposed are the support for the extended notion of federations and a broad variety of strategies to establish and maintain identity in such systems. One of the key ideas that we focus on to prevent identity theft is the notion of multi-factor identity verification. For this purpose, we present methodologies for identity assurance and new cryptographic primitives that allow privacy preserving multi-factor identity verification. We extend the basic approach with the use of biometrics by devising new techniques for biometric key generation. Further, to make decisions based on the history of activities of a user in a federation we provide methodologies to capture and use history-based identity information. We also show how this information can be used with mobile devices.

This chapter is organized as follows. In Section 6.1, we explore the most relevant federated digital identity management initiatives, describing the security and privacy features relevant to the identity theft problem. Several cryptographic techniques have been proposed for privacy preserving identity verification in distributed systems. Therefore, in Section 6.2 we compare our work in Chapter 3 to some known cryptographic schemes namely anonymous credential, identity based encryptions and signatures with zero-knowledge proofs. As the contribution of the present thesis requires the interplay of different technologies including biometric verification systems and biometric key generation techniques, in Section 6.3 we provide background information of existing biometric verification schemes and upcoming biometric key generation schemes and compare them with our work in Chapter 4. Finally in Sections 6.4 and 6.5 we provide related work in history-based trust establishment and management of identity data on mobile devices respectively, and how they compare to our techniques presented in Chapter 5. The aim of these sections is to provide state-of-the-

art in the corresponding areas to show how the integration has been exploited to provide a comprehensive solution.

6.1 Identity Management Initiatives

Identity management is being investigated extensively in the corporate world and several standardization initiatives for identity federation are being developed. A summary of some of the most significant ongoing projects are summarized in Table 6.1. In this section we first analyze the Liberty Alliance [6] (LA) and WS-Federation [17] which are the two most significant approaches. Then, we overview other relevant approaches, such as Shibboleth [13] and Microsoft CardSpace [14].

The multi-national, multi-industry Liberty Alliance (LA) [6] consortium is collaboratively developing a set of open standards for federated network identity. LA's objectives are twofold. One goal is to establish a standardized, multi-vendor, web-based single sign-on with federated identities. A second goal, which raises a number of interesting technical challenges to be achieved, is to enable organizations to maintain and manage their customer identity data without third-party participation. LA's specifications build on the Open Standard Security Assertion Markup Language [71], an XML-based security standard that provides a way of exchanging principals[1] authentication information.

LA has defined technology specifications based on three building blocks; the ID-FF (Identity Federation Framework), the ID-WSF (Identity Web Services Framework) and the ID-SIS (Identity Service Interface Specifications). ID-FF defines a framework for federating identities and a mechanism for single sign-on [128] (SSO) in a federated manner. Principals' accounts are distributed and maintained at each service site. To federate these accounts while ensuring principals' privacy, the IdP and other SPs establish a pseudorandom identifier that is associated with a real name identifier at each site. The process of federating two local identities for a principal between providers is triggered by the principal with the consent of the providers - this allows each provider to map the established

[1]Principals or users are digital representation of real world individuals in a federated IdM system (See Chapter 2 Section 2.1.1).

pseudonym into their local account identifiers. When an authentication of a principal is requested by a given SP, the IdP authenticates that principal and then issues an authentication assertion. If the IdP has already authenticated a principal, then it directly issues an assertion without requiring the principal's participation. Each SP validates the assertion issued from the IdP, and determines whether or not it should be accepted. As the IdP can issue multiple assertions to different SPs based on a single authentication action by the principal, the principal is able to sign-on to these other service sites without needing to be re-authenticated at each service site. ID-FF defines how data must be exchanged between IdPs and SPs.

ID-WSF (Identity Web Services Framework) defines a framework for web services that allows providers to share principals' attributes in a permission-based manner and to create, discover and request identity services. It also supports discovery of services and security mechanisms to transmit messages.

ID-SIS (Identity Service Interface Specifications) defines service interfaces for each identity-based web service so that providers can exchange different aspects of identity (i.e., a principal's profile) in an interoperable manner. Examples of ID-SIS services include: personal information request, geo-location services and directory services. Furthermore, LA specifies various federated identity trust models; one of which is *circles of trust*. A circle of trust is formed by federating SPs and IdPs that have business relationships and with whom principals can transact business in a secure and seamless environment.

WS-Federation is a collaborated effort of BEA Systems, BMC Software, CA Inc., IBM, Layer 7 Technologies, Microsoft, Novell and VeriSign. It is often abbreviated as WS-*. It is integrated into a series of other web services specifications such as WS-Trust [129] and WS-Security [130]. WS-Federation describes how to manage and broker the trust relationships in a heterogeneous federated environment, including support for federated identities, sharing of attributes, and management of pseudonyms. In WS-Federation, the principal obtains security tokens from its IdP and can pass them to SPs to get access to resources. The defined Web browser mechanisms allow the expressiveness provided by WS-Trust, WS-Policy, and other WS-* mechanisms to be leveraged in Web browser environments.

WS-Federation framework allows attributes to be brokered from identity and security token issuers to services and other relying parties without requiring principals intervention.

WS-Federation has been created with goal of standardizing the way companies share principals and machine identities among disparate authentication and authorization systems spread across corporate boundaries. This translates in mechanisms and specification to enable federation of identity attributes, authentication, and authorization information, but it does *not* include trust establishment/verification protocols.

The common objectives of both LA and WS-* proposals have been primarily to reduce the number of user-business interaction and exchange of information such that critical private information is used only by appropriate parties. Both approaches make principals' information available to the SPs on demand, online and with low delay. Thus, principals' data is more up-to-date and consistent compared to the case where each principal has to maintain its data in multiple places. Both reduce costs and redundancy because organizations do not have to acquire, store and maintain authorization information about all their partners' users anymore. Also, both try to preserve privacy, as only data required to use a service is transmitted to a business partner.

As compared to LA and WS-* that use PKI for principals authentication, we show how we can also leverage the SSO ID for establishing different types of identity attributes as detailed in Chapter 2. This enables privacy and adds flexibility to the identity system. We also address other security issues that remain open in both approaches. For example, almost the only security issue considered in the standards is communication security. In our work, we investigate a critical component regarding how identity verification is done in the various stages of the identity lifecycle to provide high assurance enrollment, management and use of principals identity information. Regarding privacy, as the systems are primarily provider centric, the principals must be able to regulate which information about them is allowed to be sent to which providers. However, there are no concrete definitions of such attribute release policies (ARP's) in the specifications. Our approach instead is user centric where the principal is in control when its identity attributes are used and for what purpose. Our identity protection and verification protocols can be used within the LA and WS-*

federation frameworks to ensure specific security and privacy properties as presented in Chapter 1.

Shibboleth [13] is an initiative by universities that are members of Internet2 [131]. It is a standards-based, open source middleware architecture providing both intra-domain and inter-domain SSO capability. Shibboleth implements the OASIS Security Assertion Markup Language (SAML) standard specification, and is interoperable with Microsoft's Active Directory Federation Services (ADFS) [11]. A Shibboleth federation is an agreement among resource (service) providers and institutions (IdPs). For sharing to occur, all parties need to agree on a common set of acceptable authorization attributes for their principals, and a schema to describe them. Principals' attributes are stored at the IdPs of the principals' home institution. Attributes can be encoded in Java or pulled from directories and databases. Standard X.520 [132] attributes are most commonly used, but new attributes can be arbitrarily defined as long as they are understood and interpreted similarly by the IdP and SP in the transaction.

A key aspect of Shibboleth is the emphasis on principals' privacy. The SP releases principal's attributes on the basis of the *Attribute Release Policies* (ARP's) specified by that principal. ARP's dictate the conditions according to which attributes can be released. As such, the target SP only knows the attributes and information necessary to perform an access control decision, protecting principals' anonymity in cases where their unique identity is not required. This allows flexibility about how the principal attributes are released. Our approach to identity verification can be applied in the context of Shibboleth where the principal provides multi-factor proofs of identity of SIT attributes. Here a key difference from the provider centric approach of Shibboleth would be that the principal would need to be involved when the identity proof is created and such information cannot be replayed even if the identity attributes at the IdP are compromised.

CardSpace [14] is part of Microsoft's implementation of an identity metasystem based on standard WS-* security protocols (including WS-Security, WS-Secure Conversation, WS-SecurityPolicy, WS-MetadataExchange and WS-Trust). CardSpace functions as a "digital wallet" that stores pointers to digital identifiers of a principal at various IdPs, and

provides a unified interface for choosing the identity for a particular transaction, such as logging in to a web site or accessing some web service. The CardSpace user interface enables principals to create personal cards (also known as self-issued cards) associated with a limited set of identity attributes. As a result of the selection, the CardSpace process contacts the selected IdP, and obtains an IdP signed XML document that contains the requested identity information.

Similar to the CardSpace implementation of digital wallet, in our implementation we consider an "identity wallet." Differently from their digital wallet, in our case the identity wallet contains the cryptographic secrets and commitments along with the other information related to the principals IdR stored at the registrar. The identity wallet can be used *without* contacting the registrar, contrary to the requirement in CardSpace where the IdP needs to be contacted each time an identity attribute needs to be used. This is because the principal can create the ZKPK revealing the minimal information as needed by the SP. This proof is dependent on the commitments that are signed by the registrar and stored in the identity wallet. If the SPs need to check for revocation of the signed commitments, then the revocation mechanisms described in Chapter 2 are used. Our approach also prevents against replay assuming not all the user's secrets involved in the proof are compromised. In the case of CardSpace, it is possible that if the attacker colludes with the IdP, it can retrieve and misuse the honest principal's identity attributes.

Table 6.1 presents a short summary of the above mentioned initiatives as well of other relevant projects in the area of digital identity management.

Concerning the problem of identity theft, LA, the Shibboleth project and other organizations such as Better Business Bureau and Federal Trade Commission have initiated efforts aiming at educating consumers and preventing identity theft. A LA paper [133] points out that the use of SSO in federations helps reduce ID theft by reducing the number of login names and passwords. The paper also discusses how attribute sharing in a federation inherently prevents from theft of identity attributes *"by controlling the scope of access to participating websites, by enabling consent-driven, secure, cross-domain transmission of a user's personal information.*" LA tries to mitigate ID theft attacks by having the or-

Table 6.1 Federated identity management projects and initiatives.

Liberty Alliance [6]	The Liberty Alliance is a consortium of over 150 companies that develops specifications for federated identity management. It released the first version of its Liberty Web Services Framework in 2003 which allowed single sign-on and account linking between trusted partners.
WS-Federation [17]	In April 2002, Microsoft and IBM published a joint whitepaper outlining a roadmap for developing a set of Web service security specifications. Their first jointly-developed specification, WS-Security, offers a mechanism for attaching security tokens to messages, including tokens related to identity.
Shibboleth [13]	Shibboleth is standards-based, open source middleware which provides SSO across or within organizational boundaries. It allows sites to make informed authorization decisions for individual access of protected online resources in a privacy-preserving manner.
Microsoft CardSpace [14]	Windows CardSpace, formerly known as InfoCard, is a framework developed by Microsoft which securely stores pointers to digital identities of a person, and provides a unified interface for choosing the identity for a particular transaction, such as logging in to a website or accessing web service.
OpenID [16]	OpenID is a decentralized identity system, in which any individuals online identity is given by URL (such as for a blog or a home page) and can be verified by any server running the protocol.

ganizations in the federation adopt LA standards of security, by distributing information to avoid single point of failure, by having access control on these attributes based on user preferences, and by coordinating response to incidents and frauds. There are several ongoing projects to achieve such goals. However no identity verification protocols to mitigate ID theft have been developed. In particular, none of the proposed techniques in LA takes into account the case of malicious providers. Also the LA approach does not address the problem of impersonation attacks where an attacker attempts to claim compromised identifiers to as its own. Our solution not only exploits the advantages of a federation, as the general usage case, but extends it further with the concept of SIT attributes and efficient multi-factor verification techniques.

As a final remark, from a systems standpoint, IdM systems are realized by various protocols and technologies. The basic technology related to authentication techniques such as security tokens, public keys and certificates [36,37] are part of an IdM system and are used in various steps of the IdM protocols. For example Windows CardSpace based IdM systems can employ various authentication techniques and security tokens that are requested from an IdP and passed on to a SP. More specifically, CardSpace can work with any digital identity system, using any type of security token, including simple usernames, X.509 certificates [50], Kerberos tickets [134], SAML tokens [71], and so forth. IdM protocols are typically built to allow different authentication protocols and other technologies to interoperate. Upcoming standards such as WS-* and SAML are used to exchange messages that encode the various tokens needed in various authentication protocols. In the case of CardSpace, a SPs policy is described using WS-SecurityPolicy, that policy is retrieved using WS-MetadataExchange, a security token is acquired using WS-Trust, and that token is conveyed to the SP using WS-Security.

Even though the security tokens in traditional IdM systems have typically been focused on conveying only authentication information, it is important to note that the notion of digital identity is more general as described in Section 1.1.1 of Chapter 1. Security tokens in emerging IdM systems convey various types of identity information as needed in the transaction. This use of digital identities can now become as broadly useful in the networked world as are the many identifiers we use in the real world [14].

6.2 Cryptographic Schemes

Several cryptographic schemes relevant to IdM systems and protocols have been investigated. In this section, we focus on those that are closely related to ours. We describe the work on anonymous credentials, followed by identity based encryption techniques and finally the work on signatures with zero knowledge proof.

6.2.1 Anonymous Credentials

There are few emerging IdM initiatives that are based on the notion of anonymous credentials [15, 41, 135]. In anonymous credential systems, organizations know the principals only by pseudonyms. Different pseudonyms of the same principal cannot be linked. Yet, an organization can issue a credential to a pseudonym, and the corresponding principal can prove possession of this credential to another organization (who knows it by a different pseudonym), without revealing anything more than the fact that it owns such a credential [136]. The main idea regarding use of pseudonyms in current IdM systems [6, 13, 17] is in that "the IdP generates an *opaque handle* that serves as the name identifier the SP and the IdP use in referring to the user when communicating with each other" [137]. Rudimentary non-linkability is achieved, as an outside observer cannot infer any information about the actual user based on the random session based opaque handles. The first approach that proposed the use of pseudonyms was provided by Chaum [42]; the key idea was to use one time pseudonyms for a series of transactions to provide unlinkability among different transactions with organizations, and at the same time transfer certified attributes among these organizations. A credential system was also employed, to ensure that only the information required for the transaction is revealed on a *need to know* basis.

Brands [47] significantly improved on Chaum's basic blind-signature based system in both the discrete log and strong RSA assumption settings. Brands credentials provided algorithms that provided privacy through selective disclosure in an unconditional security setting. Brands protocols include an efficient observer setting that involves augmenting security with a low performance smart card without compromising privacy guarantees. Brands' scheme also provides unlinkability features using single-use certificates, that is, certificates may only be used once if unlinkability is to be retained.

An anonymous credential system with multi-show unlinkability was provided by the Identity Mixer also known Idemix [15], which is based on Camenisch-Lysyanskaya signature scheme [41]. Idemix provides mechanisms for efficient *multi-show*[2] credentials and a

[2]Credentials can be used multiple times. Possession of a multi-show credential can be demonstrated an arbitrary number of times; these demonstrations cannot be linked to each other [15].

flexible scheme for issuing and revoking anonymous credentials. It also provides a mechanism for *all or nothing* sharing and a PKI-based non-transferability. These techniques were used in the direct anonymous attestation (DAA) protocol [138] to issue a certificate (attestation) to a computing platform that it is genuine. The techniques allow a platform to remotely prove to a SP that it is indeed genuine while protecting the platform users privacy. The attestation is issued to the trusted platform module (TPM) [139] embedded into the platform.

The anonymity properties of anonymous credentials are however limited to certified attributes and weak identifiers. Therefore anonymous credentials may not be sufficient for identity verification in several real applications because this would rely on use of strong identifiers. We differ from these approaches in that we do not hide the user identity even if we protect its identity attributes. More specifically, we do not only protect user privacy but also protect the use of its strong identifiers without requiring anonymity. Table 6.2 presents comparison between various anonymous credential schemes [15, 41, 47] and our proposed VeryIDX approach according to relevant criteria. In particular, identity theft prevention approach as provided in our work is through multi-factor identity verification are not covered by anonymous credential schemes. Other additional mechanisms such as assurance evaluation and detection of duplicate registration of strong identifiers also help in preventing identity theft as elaborated in our work that are also not part of anonymous credential schemes. Using anonymous credentials within the VeryIDX framework and protocols can provide in achieving additional security and privacy properties specific to the anonymous credential schemes employed.

6.2.2 Identity Based Encryption

The notion of Identity Base Encryption (IBE) was first introduced and defined by Shamir in 1984 [118] and then extended by several other researchers [55,140,141]. An IBE scheme is a public-key cryptosystem in which any string is a valid public key. In particular, email addresses and dates can be public keys [142]. The private key is then computed by

Table 6.2 Comparison of anonymous credential schemes and SIT attribute scheme.

Criteria	Anonymous Credential Schemes [15, 41]	**Proposed VeryIDX Scheme**
Digital Identity	Pseudonyms, PKI-based anonymous credentials, primary focus on *weak identifiers*.	PKI-based credentials, *uncertified attributes*, *SIT attributes*, primary focus on *strong identifiers*.
Privacy	Provides mechanism for minimal attributes disclosure from a user credential and provable unlinkability.	Provides mechanism for minimal attributes disclosure in the multi-factor identity verification protocols using Aggregate ZKPK. Provable unlinkability for specific protocols when strong identifiers are not required in clear.
Anonymity	Provides provable unlinkability, multi-show credentials and is mainly based on disclosure of weak identifiers.	Provides provable unlinkability for Aggregate ZKPK protocols not requiring strong identifiers in clear.
Multi-factor Identity Verification	Verification depends on a single PKI based cryptographic secret for ZKPK therefore does not provide multi-factor verification.	Verification depends on multiple (possibly non-PKI such as biometric) secrets of strong identifiers for Aggregate ZKPK resulting in multi-factor identity verification.
Duplicate detection	No mechanism to detect duplicate registration of strong identifiers.	DHT based duplicate registration detection of strong identifiers.
Confidentiality	Provides provable unlinkability, *multi show of credentials* property by which even when SPs collude no user information is leaked.	Linking of data is possible if SPs collude who have strong identifiers of a user. No identity information is leaked if even if SPs and registrars, not having the strong identifiers in clear, collude.
Integrity	Unforgeability of credentials is ensured based on the Discrete Log Problem (DLP) and strong RSA assumption.	Unforgeability of SIT attributes is based on SIT identifiers signed by registrars and Aggregate ZKPK which depends on DLP and strong RSA assumption.

a master authority in possession of the master secret, and delivered to the principal after authentication, usually via a separate channel. As a result, parties may encrypt messages or verify signatures with no prior distribution of keys to individual participants. This is

useful in cases in which pre-distribution of authenticated keys is inconvenient or not feasible because of technical constraints. However, to decrypt or sign messages, the authorized principal must obtain the appropriate private key from the private key generator. A caveat of this approach is that this private key generator must be trusted.

In the approach by Adida et al. [55] the IBE is used to define and implement a cross domain identity-based ring signatures. The ring structure of these signatures provides repudiability. With identity-based public keys, a full PKI is no longer required. Separability allows ring constructions across different identity-based master key domains. Together, these properties make signature constructions a possible solution to the email spoofing problem. Our approach greatly differs from the IBE schemes because we do not provide a mechanism to encrypt data or manage certificates. Instead we focus on providing the infrastructure and methodologies to protect the identity of a user from misuse. Typically in IBE the public information, such as the email address, is assumed to be correct and is denoted as the identity of the receiver. There is no clear methodology to verify and guarantee if this public information is correct and does belong to the intended recipient. Therefore the problem of identity theft as described in Chapter 1 is not addressed by such schemes.

6.2.3 Signatures with Zero Knowledge Proof

The work most closely related to our protocols in Chapter 3 are the cryptographic schemes proposed by Camenisch *et al.* [21]. They propose efficient protocols that allow one to prove in zero-knowledge the knowledge of a signature on a committed (or encrypted) message and to obtain a signature on a committed message. Their approach also provides a signature scheme that is based on an assumption introduced by [143] and uses bilinear maps. In Section 3.4 of Chapter 3, we show how our protocols are substantially better for the purposes of multi-factor identity verification. We combine our ZKPKs with the aggregate signature scheme presented in [22] and establish a new cryptographic primitive for aggregate proof of knowledge. Our scheme is more flexible and efficient and requires less storage than the protocols in [21]. The paper by Boneh *et al.* [22] presents several appli-

cations for aggregate signatures and proposes an efficient aggregate signature mechanism based on bilinear maps. They however, do not investigate signatures on commitments that can be used later for ZKPK protocols. Also, in our case since the signatures are aggregated by the same registrar, the aggregation and verification are more efficient. There are no other cryptographic schemes that have the same or similar functionality.

6.3 Biometric Verification Schemes

In the following we first introduce the traditional biometric matching based verification system. Then we focus on the main biometric key generation work that has been proposed in the literature.

6.3.1 Biometric Matching Based Verification Systems

Biometric verification, unlike conventional approaches, is not based on what an individual knows or possesses, but on some characteristics of the individual itself. We elaborate on the main concepts related to biometric verification in this section.

A detailed reference model for a biometric system has been developed by ISO/IEC JTC1 SC37 [4], which aides in describing the sub-processes of a biometric system. Typically there are four main subsystems in the biometric model, namely the *Data Capture, Signal Processing, Data Storage, Matching* and *Decision* subsystems.

- *Data capture subsystem:* It collects the subject's biometric data in the form of a sample that the subject has presented to the biometric sensor.

- *Signal processing subsystem:* It extracts the distinguishing features from a biometric sample to then either be stored as the reference template during registration or be matched during verification. A template is data, which represents the biometric measurement of an individual, used by a biometric system directly or indirectly for comparison against other biometric samples.

- *Data storage subsystem:* Reference templates are stored either at the server or at the client depending on the chosen architecture.

- *Matching subsystem:* It compares the features extracted from the captured biometric sample against one or more enrollment reference templates. The obtained similarity scores are then passed to the decision subsystem.

- *Decision subsystem:* It uses the similarity scores generated from one or more matching comparisons to make a decision about a verification transaction. The features are considered to match a compared template when the similarity score exceeds a specified threshold.

A biometric system typically supports two sub-processes: registration (also called enrollment), and verification. *Enrollment* is the process of capturing the features from a biometric sample provided by an individual and converting it into a template. The effectiveness of enrollment strictly depends on the quality of the data submitted along with the biometric. Thus, the enrollment process has also to ensure that the verification documents (such as passports and driver's licenses) are trustworthy so that a fake or false identity is not linked to a biometric. Additionally, no duplicate records have to be stored in the database for the same identity. This enrollment mechanism is a key aspect of biometric verification making it reliable. Enrollment is the first interaction of the user with the biometric system, and misuses of such operation can affect the quality of sample being provided by the user, which in turn affects the overall performance of the system. Once the process of registration is successfully completed, the individual can use the biometric system for verification. The *verification* is performed when the individual presents his/her biometric sample along with some other identifier which uniquely ties a template with that individual. The matching process is performed against only that template.

In traditional fingerprint based biometric verification systems [25, 26], verification is based on matching of fingerprints. One way to do the matching is to extract the minutiae points of the fingerprint and compare it against the second fingerprint template minutiae's. The effectiveness of such systems are based on evaluating error rates such as False Accept

Rate (FAR), False Reject Rate (FRR), and Equal Error Rate (EER). The processing time for the matching has shown to be efficient (0.2-0.4 seconds) for practical purposes. In our work related to biometrics as detailed in Chapter 4, there is no matching of the actual fingerprint template, therefore the efficiency of the biometric-key system is reliant primarily on the time needed to generate the biometric-key. This has been shown to be taking < 1 second with the provided hardware and software specifications, and using our key-generation techniques. Most computation is performed in the configuration phase (i.e. before enrollment), and the user specific classifier model is evaluated at the time of enrollment. Henceforth, at verification only the hashing and classification has to be performed.

6.3.2 Cryptographic Key Generation from Biometrics

Biometric based key generation has been extensively investigated in the past years. As suggested in [80] the known methods for generating cryptographic keys from biometric measurements are characterized by two stages. The first stage is when certain biometric features are examined and used to compute a bit string representing that biometric. This bit string should have uniqueness and repeatability properties, in that two different biometrics should produce different bit strings (large inter-class variation) and the same user with the same biometric should be be able to produce the same or similar bit string (small intra-class variation). This bit string in our work corresponds to the BK-HASH-VECTOR introduced in Section 4.1 of Chapter 4. The bit string is then used in the second stage to generate a unique cryptographic key with the help of stored meta data. If two instances of the bit strings are sufficiently similar then the cryptographic key generated is the same. This cryptographic key is denoted by BK-CLASS-COMB in our work. In most existing work, the second stage is independent of the biometric being used, the first is mostly specific to the biometric.

The first work in biometric key generation is because of Soutar *et al.* [76,77,144] where they describe methods for generating a repeatable cryptographic key from fingerprint using optical computing and image processing techniques. At enrollment phase, Soutar's algorithm uses image processing techniques based on correlation to create a *filter* function from

a series of fingerprint images. This filter function is combined with a random array and then an output pattern is created using an inverse Fourier transform. This output pattern is linked to a specified digital key. Then during the verification phase the digital key is retrieved by using another set of fingerprint images and the previously generated filter. The authors mainly focused on the second-stage of the key generation, which in their context is the creation of the filter function. Here the challenge is to create a filter function that is tolerant to distortions (minimal intra-class distance) and can still discriminate among different samples (maximal inter-class-distance). There are similarities of our SVD based image analysis with the Fourier analysis in the way the image vectors are produced. Similar to low-pass filtering in Fourier analysis, SVD analysis also permits filtering by concentrating on those singular vectors that have the highest singular values. We extend their main approach in how the SVD is applied and used with the SVM stage-two component. We also provide several empirical results and analysis validating our approach. In [144] the final key is independent yet linked to the biometric data, while in our approach, the final key is not only linked to the biometric data but the value of the final key depends on the value of the biometric bitstring generated in stage-one.

Most of the approaches that followed this work have a key binding aspect, in that the cryptographic key and the biometric data are monolithically bound within a cryptographic framework and it is computationally infeasible to decode the cryptographic key or the biometric template without any knowledge of the users biometric data. This differs from work as the final key is derived from the biometric data itself and cannot be pre-determined. In the former case the biometric data is *linked* with the cryptographic key, and this key cannot be retrieved without the biometric. In our work, the final value of the key that is generated is dependent both from the stored cryptographic secrets and the biometric features.

Following Soutar's work several strategies have been proposed for improving the second-stage of the key generation. Davida *et al.* [78] described a second-stage strategy using error correcting codes and how it could be used with the first-stage approaches for generating a bitstring representing iris scans [145]. The second-stage approach was significantly improved by Juels *et al.* [74, 75]. The underlying intuition behind the error correction and

similar schemes can be understood based on Shamir's secret sharing scheme [146]. Here a user can share a secret κ into n shares and any t of these shares can be combined to create the original secret. In the context of the biometric key generation, each of the n shares correspond to an element of the biometric representative string that is the resultant output of stage-one of the key generation process. As explained in [75] Shamir's secret sharing scheme cannot be used as it is for biometric data because of the noise factor inherent to biometric data. To overcome this, *error correcting codes* (ECC) is used. The Reed-Solomon error correcting codes has been viewed as the error tolerant form of Shamir's secret sharing. As such, the difficulty of Shamir's secret sharing scheme is based on the *polynomial reconstruction* problem. This problem is a special case of the Reed-Solomon list decoding problem [147]. In Juels *et al.* fuzzy vault scheme [75] using ECC the user adds spurious *chaff* points that make it infeasible for an attacker to reconstruct the polynomial representing the biometric key of the original user.

Since the introduction of the fuzzy vault scheme, several researchers have implemented it in practice [148–154]. In particular the most recent work is by Nandakumar *et al.* [154] where the fuzzy vault implementation is based on the location of minutia points in a fingerprint. They generated 128 bit keys and obtained an accuracy rate of 91% for high quality images and 82.5% for medium quality images. The FRR was approximately 7% which shows an improvement over several other implementation of this scheme (where the average FRR was from 20-30%). From the experimental point of view, we generate 134 bit keys with the accuracy of 94.96% for high quality images and 86.92% for medium quality images. The FRR was on an average 9.06%, which is comparable to the above scheme. From the algorithmic point of view, we use a similar concept of chaff points while adding spurious classes to make it difficult for the attacker to guess the correct final key. We do not use error correcting codes to retrieve the final key, but plan to investigate how they can be used while finding a list of SVM classes uniquely ordered by the confidence measures (See Chapter 4 Section 4.3.3). Comparing our approach with respect to the stage-one approaches of the various implementations of the fuzzy-vault one major difference is that their feature extraction is specific to the biometric type. In our case, we instead use image

analysis that can be used for several generic 2D biometric images like fingerprint image, iris image and face image.

Another scheme which makes use of the *polynomial reconstruction* problem in the second-stage is the scheme proposed by Monrose *et al.* that was originally used for hardening passwords using keystroke data [155] and then extended for use in cryptographic key generation from voice [80]. In this approach consider if m biometric features are recorded as a result of stage-one. It follows, when the system is initialized the main key κ and 2m shares of κ are generated using a generalized secret sharing scheme. The shares are arranged within an $m \times 2$ table such that κ can be reconstructed from any set of m shares consisting of one share from each row. The selection is based on the biometric features recorded and they show that it is computationally infeasible for an attacker to guess the right shares because of the random or spurious shares present in the table. As described in the case before where we also add spurious classes in the SVM classification model. Moreover, the features they capture in stage-one for key stroke [155] are durations and latencies and voice [80] are ceptral coefficients. For their experiments they obtained on an average about 20-30% FRR. This biometric encoding of voices is not comparable with ours as we consider different biometrics that can be represented in 2D images.

Several of the above described techniques have been recently extended in the context of BioHashing [27, 156, 157]. The ones closest to our work are the bio-hashing techniques by Goh and Ngo [81, 82] where the authors propose techniques to compute cryptographic keys from face bitmaps. *Bio-hashing* is defined as a transformation from representations that are high-dimension and high-uncertainty (example face bitmaps) to those that are low-dimension and zero-uncertainty (the derived keys). Similar to our work, the goal of using the image hashing techniques is to extract bits from face images so that all similarly looking images will produce almost the same bit sequence. However, the work mainly focuses on the first stage of biohashing and propose potential use of Shamir's secret sharing techniques [146] in the second stage.

With respect to the first stage, the authors use principal component (PCA) analysis while analyzing the images. This is similar to our use of SVD, as both SVD and PCA are

common techniques for analysis of multivariate data. There is a direct relation between PCA and SVD in the case where principal components are calculated from the covariance matrix. The right most and left most eigen vectors in Algorithms 1 and 2 of Chapter 4 which retrieved are the same as the principal components in the given context. An important capability distinguishing SVD and related methods from PCA methods is the ability of SVD to detect weak signals or patterns in the data that is important in our case as we propose to use our techniques for generic 2D biometric images. The methodologies we employ for stage-one also differ in that the BK-HASH-VECTOR output from stage-one cannot be simply distinguished using straightforward implementation of Hamming distance based analysis as proposed in [81,82]. We instead couple stage-one and stage-two with the use of SVM classifiers in stage-two that provides a way to analyze the properties such as inter and intra-class distance of the BK-HASH-VECTORS. We provide a detailed analysis of this approach.

There are other biometric cryptosystems where biometric authentication is completely decoupled from the key release mechanism. The biometric template is stored on the device and when the biometric match happens then the cryptographic key is released [158]. This approach however has several vulnerabilities and is not related to our key generation approach.

6.4 History Based Trust Management Initiatives

Transaction history-based trust establishment has been explored from different perspectives. We elaborate on three different perspectives; the *reputation systems* that rely on the history of e-commerce activities of the principals; the *transaction protocols* that ensure fair and safe transactions; the *cryptographic protocols* that ensure unforgeable receipts. Related work in each area is detailed in the following.

Reputation systems have been investigated extensively. One approach to build a reputation system is to have a distributed trust management systems [29]. The basic idea of this work is to construct hierarchical reputation systems. Principals who want to know

reputation for a specific server (seller), query a local broker. Reputation is calculated from principals' evaluation after the completion of a transaction with the server. This score is merged throughout several brokers. Note that in this framework only servers' rating of the principals is stored and not the attributes on which the score is calculated. In our system it is possible for SP to draw reputation score from principals' transaction history given principals' consent to view the receipts. Another key difference is with respect to the subject who uses reputation scores. In [29], principals, as buyers, take advantage of SPs reputation score to choose trustworthy sellers whereas in our approach, the sellers utilize principals' past transaction history to determine principals' reputation.

Another approach to reputation systems has been developed in the context of P2P networks [109]. Such an approach does not depend on the customers' evaluation of the seller. Instead it suggests new credit computation schemes of a reputation system for decentralized unstructured P2P networks such as Gnutella [159]. The proposed system has two computation schemes, namely the debit-credit reputation computation (DCRC) and the credit-only reputation computation (CORC). In P2P system, every principal shares files with other principals and get files from them. Each principal as a peer is both a client and a server in these networks. So when a principal joins the system, its machine becomes a peer (server) to others. The reputation score of a principal, as a server, is an important factor for decision-making- who to download content from. This score is raised as peers download more files from it.

One novel contribution of this work is to enable a peer to keep its reputation locally for the fast reputation retrieval. A reputation computation agent (RCA) prevents malicious reputation modification by use of a public key based mechanism. Unlike real world transactions, a sender (who shares files with others) is the one who gets receipts from receivers. Senders report those receipts to the RCA and receive the updated reputation score about themselves in return. In our approach we provide a way for both the buyer and the seller to create and submit receipts of their past transactions. In [109], the use of receipts is restricted to acquire credit from the server. By contrast, our receipts can be used as a proof of purchase for other types of transactions as well.

In the AttentionTrust [160] approach, principals install a Firefox extension to share their website access log with the SPs. This system supposes that principals are willing to share their privacy without gaining any financial benefit. The information sent to a central server and SPs may be used for customized advertisement to the users. The extension sends web page URL, title, HTTP response code and so on to the server. This is similar to ours in that principals are allowed to choose SP to share privacy information. However the principals of AttentionTrust cannot choose which information will be shared, though the principals can specify the list of websites that should not receive its data. In our case, the principals can choose which information will be revealed to which SP, at what time and for what purpose.

Transaction protocols provide mechanisms to execute price negotiation, ordering and payment procedures. For example, a transaction server named NetBill for information goods was suggested in early 90s [161]. It takes part in the payment procedure so as to allow a buyer to hide its identity from the seller and give certified receipts to the buyer. The main goal of this system is to assure a fair exchange between two parties i.e customers can read or use electronic goods only after they receive a decryption key from a merchant. The merchant sends a decryption key to the buyer only if he got payment from the user and then reports an endorsed payment order to the server. Customers receive receipts consisting of transaction result, identity, price, product ID and so on. The server signs this receipt and then transmits it to the merchant. However, it is the responsibility of buyers to manage these receipts. We also investigate fair exchange in Protocol 2 of Chapter 5. In addition, principals manage and use receipts within VeryIDX framework with assurance based on multi-factor identity verification.

Finally, cryptography-oriented approaches have been proposed that deal with history-based trust establishment. For example, Simmons and Purdy proposed ZKP of identity attributes in transaction receipts [162]. They focus on the unforgeable transaction receipts using ZKP. They use a public authentication channel to create trusted credentials. These credentials can then be used for constructing proofs. Even though receipts in their scheme can be extended for use in two-way protocol between a seller and a buyer, using this receipt

for other purposes does not seem trivial. This is because each principal's credentials are highly specialized for the proposed scheme and does not offer interoperability.

Another work related to non-repudiation in transactions is by Coffey *et al.* [163] where they propose an approach to achieve mandatory mutual non-repudiation including both mandatory proof of origin and mandatory proof of receipt. As a result, their approach ensures non-repudiation protocol and fair exchange. This research is more focused on the transaction itself rather than on the receipt management. As our system is not affected by the payment process, those techniques could be used together with the ones presented in Chapter 5.

6.5 Mobile Identity Management Initiatives

In this section we discuss the related work on using mobile phones for commerce transactions involving identity attributes and other recent developments in mobile identity management initiatives.

With the advent of high-speed data networks and feature-rich mobile devices, the concept of *mobile wallet* [164–166] has gained importance. The initial efforts of combining digital cash and mobile telephones was under two distinct projects by CWI Amsterdam. One was on mobile device authentication, the other on Chaum's online digital payment protocols [42]. In both cases the idea was to connect to the bank and payee via the mobile networks, using Global System for Mobile Communication (GSM) [167] mobile terminal as the payer's electronic wallet. Subsequently, as a part of the European CAFE e-commerce project [165] this idea was extended in a seminal work introducing the concept of wallets with observers [84], which enabled off-line digital cash and credentials to be used in commercial settings. The CAFE project developed electronic wallet technology; the transactions are performed via a short range infrared link either directly with compliant cash registers and wallets held by other individuals, or over the Internet, to other SPs. Although functionality of the CAFE wallet was never demonstrated in combination with cellular technology the project results in the significant step for the mobile wallet technol-

ogy. The functional elements of the proposed mobile wallet was comprised of an observer and a purse. The observer is trusted by the credential issuer and protects the issuers interest during off-line transactions. The observer restricts the copying and uses the credentials on behalf of the issuer. An off-line transaction in this respect is a transaction where both the credential holder (payer) and the credential verifier (payee) do not connect to any auxiliary services. The purse is owned and trusted by the payer. Our approach does not require an observer, as the integrity of the receipts is based on the signature of the registrar on the receipts. As a part of future work, the addition of the observer would be beneficial if the usage of the receipts is constrained for example by number of times of use.

The wallets with observers approach was generalized in [164]. Here the authors also exploited the on-line mobility of the user's device, and the available wireless networks. They solved the multi-issuer problem in the original approach by having the mobile keep a single access credential corresponding to an entity called a localized credential keeper at the user's (possibly remote) machine. The localized credential keeper stores all credentials issued for different services and was accessed online during the transaction. No centralized on-line server carrying sensitive personal credential information needs to be established in their approach. Our approach could be decentralized if several registrars were involved. Moreover we overcome the problem of multiple issuers of the receipts, as all receipts are signed by the registrar when they are stored in the RREC. The signature on multiple receipts can be verified using the aggregation techniques presented in Chapter 3.

A recent commercial example of a mobile wallet is the Valista mobile wallet [168], which allows functions such as secure payment transactions, personalization and user identity verification. The authors employ a provider centric approach, wherein the wallets are hosted on a server (like an IdP) and accessed from the user's mobile device. This methodology gives the IdP control over how the data is used and the security of the data and transactions. The information provided via the wallet is as per the requirement of the SP service policy. The mobile wallets proposed here are online and adopt IdM services on the high-speed networks. The wallets comply with the major security standards, such as Visas Mobile 3-D Secure and MasterCards Secure Payment Application (SPA). Moreover,

the Liberty Alliance Project [6] has taken initiatives to drive multi-device user identity that employ such wallets. An important distinction of our approach from the one presented by Valista is of user centricity. In our case the wallet is stored and secured in the device itself and does not require the IdP to be contacted each time any wallet attribute is used. We also provide ZKPK techniques to ensure minimal disclosure of the attributes, and techniques allow the individual to choose the identity information to reveal based on the SP service policies.

Other mobile identity management initiatives have gained importance with the rapid adoption of second-generation mobile telecommunication systems, leading to the growth of mobile commerce [119, 167]. Two specific factors that are critical in this domain are usability and trust. Several approaches to enable usability of the mobile devices have been proposed [123]. Trust on the device comprises several security and privacy properties such as confidentiality, integrity, user control and minimal disclosure of the identity data stored on these devices.

One approach to mobile IdM is based on the GSM [167]. GSM based IdM uses the GSM infrastructure and the Subscriber Identity Module (SIM) as the underlying platform. Using the GSM based mobile IdM has several advantages but the identity attributes managed are limited and related to the SIM-Hardware or the GSM infrastructure. Identity attributes such as those involved in current IdM systems [6, 13, 17] are not supported. Our approach could use the GSM infrastructure to provide history-based and other general identity attributes. There are also several privacy and trust issues using the proposed GSM model [167] that can be mitigated using our approach of using ZKPKs and other related techniques proposed in the thesis.

In [119] the authors propose a mobile IdM solution where they emphasize user control over the data that is published based on the services that are offered. As such, the mobile device carrying identity information would reveal the partial identity based on the context of the transaction and location. More specifically, their approach consists of three main steps. In the first step their device uses the surrounding computing and interaction environment to set a context. In the second step, the device application UI provides the user an

option to choose the appropriate partial identity that would be revealed. Finally, in the third step the individual decides which services and authentication information and protocols be combined with this partial identity. They also investigate possibilities of anonymous communications with the mobile devices under specific contexts. Similar to the given approach, ours also emphasizes user consent and control on the identity attributes disclosed to the SP. We differ by the use of aggregate ZKPKs and other verification techniques that provide additional security and privacy properties (See Chapters 3 and 5) in the resultant mobile IdM system.

7. SUMMARY AND FUTURE DIRECTIONS

Identity management systems have improved the management of identity information and user convenience, however they do not provide specific solutions to address protection of identity from threats such as identity theft and compromise of an individuals' privacy. In this thesis we presented a number of techniques that address the problem of identity verification leading to protection of identifiers against misuse. In this chapter we provide the summary followed by possible future work.

7.1 Summary

Our approach is based on the concept of privacy preserving multi-factor identity verification that consists of verifying multiple identifier claims of an individual without revealing extraneous identity information. A distinguishing feature of the approach is the use of identity protection and verification techniques in all stages of the identity life cycle. Our approach is also enhanced with the use of biometric and history-based identifiers. In particular we provide the following key contributions:

- A new cryptographic primitive referred to as *aggregate proof of knowledge* to achieve privacy preserving multi-factor verification. This primitive uses aggregate signatures on commitments that are then used for aggregate zero-knowledge proof of knowledge (ZKPK) protocols. Our cryptographic scheme is substantially better in terms of performance, flexibility and storage requirements than existing efficient ZKPK techniques that may be used to prove, under zero-knowledge, the knowledge of multiple secrets.

- Algorithms to reliably generate biometric keys from an individuals' biometric images which in turn are used to perform multi-factor identity verification using ZKPK.

Several factors, including various traditional identity attributes, can thus be used in conjunction with one or more biometrics of the individual. We also ensure security and privacy of the biometric data and show how the biometric key is not revealed even if all the data, including cryptographic secrets, stored at the client machine is compromised. We provide an empirical evaluation of our techniques using biometric images of individuals for different types of biometrics; namely fingerprint, iris and face. As compared to related work, our algorithms perform better in terms of accuracy, false rejection rate and false acceptance rate. Our approach is also novel in terms of how the key is generated and used in the system. More specifically, we do not use biometric keys directly and instead use them to create biometric commitments that are used in the aggregate ZKPKs.

- A series of protocols for the establishment and management of individuals' transaction history-based identifiers encoded as receipts from e-commerce transactions. These receipt protocols satisfy the security and privacy requirements related to management of electronic receipts. We show how the user's receipt protocols can be employed in the context of mobile phones. In particular we provide techniques for managing the portable identity information on such devices, and using them at physical locations of the service providers. We provide a prototype implementation and performance analysis of the key protocols on the web and mobile phone settings.

7.2 Future Directions

An important aspect to explore as a part of future work is the service provider and user acceptance of the concept of multi-factor proofs presented in this dissertation. The study would include surveying verification policies that use the concept of proof of identity versus the traditional attributes in clear. Practical concerns of the use of such proofs in an identity management system that have additional compliance, legal and business requirements also need to be considered. Moreover, methodologies for usable yet secure management of

secrets associated with the secured from identity theft identifiers, at the user end, need to be investigated.

Another critical aspect not considered in the dissertation is the concept verifying negative claims. In our approach there are various requests and queries issued against the user's identity attributes that correspond to positive claims. When a SP requests an attribute, the user provides it with proof of ownership of that attribute using multi-factor proofs. Based on this, one can build additional types of queries employing other complicated technologies to infer negative claims. For example, consider a case when a given SP needs to verify that a user does *not* have a criminal record. The SP may have policies that use the domain knowledge and other additional sources regarding criminal records to collect positive claims about the individual. In this case, positive claims in the form of certificate(s) of clearance from the police department of the users state(s) of residence in the last 5 years may be needed to satisfy the SP policy. The provided user attributes can then be evaluated using an inference engine at the SP to have confidence about the truth of the negative claim. Other complex procedures can be used, utilizing the semantics of the attributes along with the policies associated with the use of such attributes.

In the rest of this section we present other applications and future directions of the various techniques presented in the thesis.

7.2.1 Aggregate Zero Knowledge Proofs

The aggregate ZKPK primitive is useful in other scenarios where a large number of proofs and signatures need to be transmitted. An example is in the case of distributed computing applications that are used to solve difficult computational tasks [169, 170]. One well known application is the SETI@Home [169] project that uses free cycles of Internet-connected computers to analyze radio telescope data in the Search for Extraterrestrial Intelligence. In such distributed applications there exists a supervisor who splits the main job into tasks executed by many participants. One main concern for distributed computation applications is the honesty of participants. Several techniques have been proposed to mit-

igate against dishonest participants [171]. One additional methodology could be to verify identity attributes of the participants, to evaluate the trustworthiness of the computations performed by them. Such verification could employ aggregate ZKPK to ensure efficiency, with the increased number of signed proofs required in the system.

The aggregate ZKPK protocols presented can be extended in several directions as a part of future work. The protocols presented in Chapter 3 are two-party protocols. One extension would be to consider multi-party [172, 173] aggregate ZKPK where n players compute their proofs separately and aggregate them in a way such that the privacy of their inputs is not compromised and the verification is successful only when each input provided by the parties is correct. Another extension would be to consider (k, n) threshold aggregate ZKPK schemes where if k of the inputs involved in the construction of the aggregated ZKPK is correct, then the verification is successful. Note that this is different from threshold signature based zero knowledge schemes [174] where the secret shares are combined in zero-knowledge using polynomial interpolation.

7.2.2 Biometric Key Generation

The biometric keys generated, using techniques presented in Chapter 4, can be useful in several other applications including access control, computer login and encrypting digital information. Biometrics are increasingly being included in identification cards of individuals [175] where the biometric templates are stored in the card itself. A possible alternative, to be explored as future work, would be to instead store the biometric commitment and the meta data required to re-generate the biometric key which in turn is used for verification. In this way, the privacy of the biometric would be preserved and the verification would satisfy the additional security properties as described in Chapter 4.

The biometric key generation algorithms presented can also be extended further as a part of future work. One extension would be to include the use of error correction codes (ECC) such as Reed-Solomon codes [150] while creating the final biometric key. Another extension would be to investigate how SVM classification itself can improve the biometric

hash vector classification and the addition of spurious classes in the model as described in Chapter 4 Section 4.3.

7.2.3 History Based Protocols

As mentioned in Chapter 5, history-based protocols can be used extensively in the context of reputation systems. There are several aspects that can be extended as a part of future work, including usability studies [123, 176] that would analyze how the individuals use the e-receipts in e-commerce transactions and in the context of portable receipts on mobile devices. Another important aspect would be securing the cryptographic secrets on such devices using techniques such as Shamir's Secret Sharing [146].

A significant advantage of our VeryIDX framework lies in the possibility for the registrar and for the SP to cooperate (by exchanging messages not part of the receipt management and usage protocols) to promptly detect possible frauds. Anomalous behavior can be detected by peers of the federation that exchange messages upon identification of fraudulent action. Frauds in the given context occur when users dishonestly use receipts with the intention of obtaining services for which they are not qualified. Fraud detection is particularly challenging as the attacker is not an outsider, but a known user who misuses its rights without breaking the protocol rules. Fraud detection relies on analysis of logs that collect history of transactions of individuals. Logs are common practices of business oriented transactions. The use of receipts and receipt protocols, would enable privacy preserving logs as a side affect of using ZKPKs. The SPs do not gather extraneous information about users attributes which could be profiled in a way to compromise user privacy. As a part of future work, cases can be considered where either a SP or a registrar suspects or detects a misuse, raises alarms to the cooperating entities and informs them about the anomaly. These methodologies may lead to possible solutions to balance the profiling of individuals' activities and preserving privacy in identity management systems.

7.3 Advantages

Our approach has several advantages as summarized below:

- Privacy of individuals is preserved, as minimal information is released, both in the registration and the usage phase. Individuals only register the identifiers they are willing to commit. At the time of usage, the actual values of identifiers are revealed only if required for obtaining the service. Additional proofs of identity can be provided by the individuals without revealing the actual values of identity attributes. The verification methods are efficient, because individuals can satisfy SPs multiple identifier verification requirements by disclosing a single piece of information. Because of the aggregate ZKPK protocol, efficiency is ensured even if proofs of multiple identifiers are required.

- The federation protocols are secure with respect to the basic security and privacy properties described in this section. Even if some information about individual identifiers is leaked to an adversary, the adversary is not able to use it for obtaining any service in the federation. The main effort required by an individual is when it first establishes identity proofs. Once this bootstrapping part is completed, the operations needed from the individual are minimal. The protocol proofs required for verification may be implemented without requiring any human intervention if the secrets are stored in tamper proof hardware.

- Our approach makes it possible to maintain consistency in a federation with respect to two well known invariants of individuals identifiers. First, strong identifiers are generally unique, unless proved otherwise by the owners. The second invariant is related to the fact that several strong identifiers of an individual have some common weak identifiers associated with them. The two invariants cover the common understanding of the notion of strong identifiers.

- Biometric identifiers are supported. The introduction of biometric verification into a framework for the verification of identity attributes is novel and will result in ad-

vances to the state-of-art with respect to the integration of cryptographic protocols and biometric data in IdM systems.

- History based identity attributes are supported. They provide a way to use individuals online activity to generate reliable identity information which can be managed and used as any other identity attributes to evaluate reputation and other trust relationship based related properties.

- The approach supports portable identifiers and their usage with mobile devices such as cellular phones. Several aspects relevant to such devices with respect to the security and resource usage are investigated.

7.4 Conclusion

This thesis demonstrates several aspects of digital identity protection and the effectiveness of our privacy preserving multi-factor identity verification solution in an identity management framework. We have established new techniques for cryptographic computations and use of biometric and history-based identifiers in such a system. Our solution is a significant advancement in the protection of identity attributes in identity management systems. Moreover, as described above our techniques can be applied in broader contexts, and have considerable scope for future work.

LIST OF REFERENCES

LIST OF REFERENCES

[1] M. Scannapieco, P. Missier, and C. Batini, "Data quality at a glance," *Datenbank-Spektrum*, vol. 14, pp. 6–14, 2005.

[2] A. I. Anton, "Testimony before the house committee on ways and means subcommittee on social on protecting the privacy of social security number from identity theft," *US Public Policy Committee of the Association for Computing Machinery*, vol. 1, pp. 1–19, June 2007.

[3] D. Woodruff and J. Staddon, "Private inference control," in *CCS '04: Proceedings of the 11th ACM Conference on Computer and Communications Security*, (New York, NY, USA), pp. 188–197, ACM Press, 2004.

[4] SC 37 Secretariat, "Text of fcd 19795-2, biometric performance testing and reporting - part 2: Testing methodologies for technology and scenario evaluation," in *ISO/IEC JTC 1/SC 37*, ANSI, 2006.

[5] "Consumer fraud and identity theft complaint data, January-December, 2005." `http://www.consumer.gov/sentinel/pubs/Top10Fraud2005.pdf`, 2006.

[6] "Liberty alliance project." `http://www.projectliberty.org`.

[7] "George W. Bush, national consumer protection week proclamation." `http://www.whitehouse.gov/news/releases/2002/02/20020204-8.html`, 2002.

[8] "Federal Trade Commission fact sheet, Aberdeen group, identity theft: A $2 trillion criminal industry in 2005." `http://www.ustreas.gov/offices/domestic-finance/financial-institution/cip/pdf/identity-theft-fact-sheet.pdf`.

[9] M. Topkara, A. Kamra, M. J. Atallah, and C. Nita-Rotaru, "ViWiD: Visible watermarking based defense against phishing," in *International Workshop of Digital Watermarking (IWDW)*, vol. 3710, pp. 470–483, 2005.

[10] R. Dhamija and J. D. Tygar, "The battle against phishing: Dynamic security skins," in *SOUPS '05: Proceedings of the 2005 Symposium on Usable Privacy and Security*, (New York, NY, USA), pp. 77–88, ACM Press, 2005.

[11] A. G. Lowe-Norris, *Windows 2000 Active Directory*. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2000. Editor Robert Denn.

[12] D. P. Kormann and A. D. Rubin, "Risks of the passport single sign-on protocol," in *Proceedings of the 9th international World Wide Web Conference on Computer networks : International Journal of Computer and Telecommunications Networking*, (Amsterdam, The Netherlands), pp. 51–58, North-Holland Publishing Co., 2000.

[13] "Shibboleth, Internet2." http://shibboleth.internet2.edu.

[14] "Windows CardSpace." http://cardspace.netfx3.com/.

[15] J. Camenisch and E. V. Herreweghen, "Design and implementation of the idemix anonymous credential system," in *CCS '02: Proceedings of the 9th ACM Conference on Computer and Communications Security*, (New York, NY, USA), pp. 21–30, ACM Press, 2002.

[16] "OpenID." http://openid.net.

[17] S. Overhage and P. Thomas, "WS-Specification: Specifying web services using UDDI improvements," in *Revised Papers from the NODe 2002 Web and Database-Related Workshops on Web, Web-Services, and Database Systems*, (London, UK), pp. 100–119, Springer-Verlag, 2003.

[18] USACM The Public Policy Committee of ACM, "Understanding identity and identification." http://usacm.acm.org/usacm/Issues/identity.pdf, January 2007.

[19] J. H. Salzer and M. D. Schroeder, "The protection of information in computer systems," in *Proceedings of IEEE*, vol. 63, pp. 1278–1308, 1975.

[20] U. Fiege, A. Fiat, and A. Shamir, "Zero knowledge proofs of identity," in *STOC '87: Proceedings of the nineteenth annual ACM conference on Theory of computing*, (New York, NY, USA), pp. 210–217, ACM Press, 1987.

[21] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Advances in Cryptology – CRYPTO '04*, 2004.

[22] D. Boneh, C. Gentry, H. Shacham, and B. Lynn, "Aggregate and verifiably encrypted signatures from bilinear maps," in *Proceedings of Advances is Cryptology – Eurocrypt'03, LNCS. Springer-Verlag, 2003.*, 2003.

[23] G. S. Manku, *Dipsea: a modular distributed hash table*. PhD thesis, Stanford University, 2004. Adviser-Rajeev Motwani.

[24] G. S. Manku, "Balanced binary trees for id management and load balance in distributed hash tables," in *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, (New York, NY, USA), pp. 197–205, ACM Press, 2004.

[25] A. Jain and L. Hong, "On-line fingerprint verification," *ICPR'96: International Conference on Pattern Recognition*, vol. 03, p. 596, 1996.

[26] N. K. Ratha, K. Karu, S. Chen, and A. K. Jain, "A real-time matching system for large fingerprint databases," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 799–813, 1996.

[27] T. Connie, A. Teoh, M. Goh, and D. Ngo, "Palmhashing: A novel approach for cancelable biometrics," *Information Processing Letters*, vol. 93, no. 1, pp. 1–5, 2005.

[28] K.-S. Goh, E. Chang, and K.-T. Cheng, "Support vector machine pairwise classifiers with error reduction for image classification," in *MULTIMEDIA '01: Proceedings of the 2001 ACM workshops on Multimedia*, (New York, NY, USA), pp. 32–37, ACM Press, 2001.

[29] K.-J. Lin, H. Lu, T. Yu, and C. en Tai, "A reputation and trust management broker framework for web applications," in *EEE '05: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service*, (Washington, DC, USA), pp. 262–269, IEEE Computer Society, 2005.

[30] "Near field communication forum." `http://www.nfc-forum.org`.

[31] E. Norlin and A. Durand, "Whitepaper on towards federated identity management." `www.durand.com/ping/FIM_Whitepaper.pdf`, 2002.

[32] A. Whitten and J. D. Tygar, "Why Johnny can't encrypt: A usability evaluation of PGP 5.0," in *8th USENIX Security Symposium*, 1999.

[33] E. Bertino, E. Ferrari, and A. C. Squicciarini, "Trust-χ: A Peer-to-Peer Framework for Trust Establishment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 827– 842, July 2004.

[34] T. Yu, M. Winslett, and K. E. Seamons, "Interoperable strategies in automated trust negotiation," in *CCS '01: Proceedings of the 8th ACM Conference on Computer and Communications Security*, (New York, NY, USA), pp. 146–155, ACM Press, 2001.

[35] "Organization for the Advancement of Structured Information Standards)." `http://www.oasis-open.org/home/index.php`.

[36] C. Adams and S. Farrell, "Internet X.509 public key infrastructure certificate management protocols," 1999.

[37] S. Kent, "Privacy enhancement for Internet electronic mail: Part ii: Certificate-based key management." RFC 1422 (Proposed Standard), Feb. 1993.

[38] "Electronic authentication partnership." `http://eap.projectliberty.org/`.

[39] W. E. Burr, D. F. Dodson, and W. T. Polk, "Electronic authentication guideline: Recommendations of the national institute of standards and technology," in *NIST SP800-63*, p. 64, NIST, 2006.

[40] H. Buerk and A. Pfitzmann, "Value exchange systems enabling security and unobservability," *Computer and Security*, vol. 9, no. 9, pp. 715–721, 1990.

[41] J. Camenisch and A. Lysyanskaya, "Efficient non-transferable anonymous multishow credential system with optional anonymity revocation," in *Advances in Cryptology — EUROCRYPT 2001* (B. Pfitzmann, ed.), vol. 2045 of *Lecture Notes in Computer Science*, pp. 93–118, Springer Verlag, 2001.

[42] D. Chaum, "Security without identification: transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[43] A. Buldas, P. Laud, and H. Lipmaa, "Accountable certificate management using undeniable attestations," in *CCS '00: Proceedings of the 7th ACM Conference on Computer and Communications Security*, (New York, NY, USA), pp. 9–17, ACM Press, 2000.

[44] I. Gassko, P. Gemmell, and P. D. MacKenzie, "Efficient and fresh certification," in *PKC '00: Proceedings of the Third International Workshop on Practice and Theory in Public Key Cryptography*, (London, UK), pp. 342–353, Springer-Verlag, 2000.

[45] A. Bhargav-Spantzel, A. C. Squicciarini, R. Xue, and E. Bertino, "Practical identity theft prevention using aggregated proof of knowledge," tech. rep., CS Department, 2006. CERIAS TR 2006-26.

[46] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, (London, UK), pp. 129–140, Springer-Verlag, 1992.

[47] S. Brands, "Untraceable off-line cash in wallet with observers," in *CRYPTO '93: Proceedings of the 13th annual international cryptology conference on Advances in cryptology*, (New York, NY, USA), pp. 302–318, Springer-Verlag New York, Inc., 1994.

[48] J. E. Holt and K. E. Seamons, "Selective disclosure blinded credential sets," in *Cryptology ePrint Archive, Report 2002/151*, 2002. `http://eprint.iacr.org/`.

[49] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, Inc., 1996.

[50] R. Housley, W. Polk, W. Ford, and D. Solo, "Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile," 2002.

[51] P. Wohlmacher, "Digital certificates: a survey of revocation methods," in *MULTIMEDIA '00: Proceedings of the 2000 ACM Workshops on Multimedia*, (New York, NY, USA), pp. 111–114, ACM Press, 2000.

[52] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 Internet public key infrastructure online certificate status protocol - OCSP," 1999.

[53] J. H. Choi, S. S. Lim, and K. Zeilenga, "A new on-line certificate validation method for improved security, scalability, and interoperability," in *6th IEEE Information Assurance Workshop*, 2005.

[54] G. Goth, "Phishing attacks rising, but dollar losses down," *IEEE Security and Privacy*, vol. 3, no. 1, p. 8, 2005.

[55] B. Adida, S. Hohenberger, and R. L. Rivest, "Separable identity-based ring signatures: Theoretical foundations for fighting phishing attacks"," in *Proceedings of DIMACS Workshop on Theft in E-Commerce: Content, Identity, and Service*, 2005.

[56] V. V. Prakash and A. O'Donnell, "Fighting spam with reputation systems," *Queue*, vol. 3, no. 9, pp. 36–41, 2005.

[57] "Microsoft Windows 2000 server public key interoperability." `http://www.alw.nih.gov/pki/`.

[58] S. Foley, L. Gong, and X. Qian, "A security model of dynamic labeling providing a tiered approach to verification," in *SP '96: Proceedings of the 1996 IEEE Symposium on Security and Privacy*, (Washington, DC, USA), p. 142, IEEE Computer Society, 1996.

[59] R. R. Henning, "Security service level agreements: quantifiable security for the enterprise?," in *NSPW '99: Proceedings of the 1999 workshop on New security paradigms*, (New York, NY, USA), pp. 54–60, ACM, 2000.

[60] J. Camenisch and A. Lysyanskaya, "A signature scheme with efficient protocols," in *International Conference on Security in Communication Networks – SCN*, vol. 2576 of *Lecture Notes in Computer Science*, pp. 268–289, Springer Verlag, 2002.

[61] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *Advances in Cryptology – CRYPTO '97*, pp. 410–424, 1997.

[62] A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," in *Proceedings on Advances in cryptology—CRYPTO '86*, (London, UK), pp. 186–194, Springer-Verlag, 1987.

[63] C. P. Schnorr, "Efficient identification and signatures for smart cards," in *EUROCRYPT '89: Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, (New York, NY, USA), pp. 688–689, Springer-Verlag New York, Inc., 1990.

[64] D. Chaum, "Zero-knowledge undeniable signatures (extended abstract)," in *EUROCRYPT '90: Proceedings of the workshop on the theory and application of cryptographic techniques on Advances in cryptology*, (New York, NY, USA), pp. 458–464, Springer-Verlag New York, Inc., 1991.

[65] C. Cachin, K. Kursawe, A. Lysyanskaya, and R. Strobl, "Asynchronous verifiable secret sharing and proactive cryptosystems," in *CCS '02: Proceedings of the 9th ACM Conference on Computer and Communications Security*, (New York, NY, USA), pp. 88–97, ACM Press, 2002.

[66] J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," in *CCS '05: Proceedings of the 12th ACM Conference on Computer and Communications Security*, (New York, NY, USA), pp. 46–57, ACM Press, 2005.

[67] B. Pfitzmann and M. Waidner, "Strong loss tolerance of electronic coin systems," *ACM Transactions Computer Systems*, vol. 15, no. 2, pp. 194–213, 1997.

[68] I. Damgard and E. Fujisaki, "A statistically-hiding integer commitment scheme based on groups with hidden order," in *ASIACRYPT '02: Proceedings of the 8th International Conference on the Theory and Application of Cryptology and Information Security*, (London, UK), pp. 125–142, Springer-Verlag, 2002.

[69] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *Lecture Notes in Computer Science*, vol. 2248, p. 514, 2001.

[70] I. F. Blake, V. K. Murty, and G. Xu, "Refinements of miller's algorithm for computing the weil/tate pairing," *Journal of Algorithms*, vol. 58, no. 2, pp. 134–149, 2006.

[71] "Security assertion markup language specification set." `http://www.oasis-open.org/committees/security`.

[72] B. Tan and S. Schuckers, "Liveness detection for fingerprint scanners based on the statistics of wavelet signal processing," in *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, vol. 0, (Los Alamitos, CA, USA), p. 26, IEEE Computer Society, 2006.

[73] H. Kang, B. Lee, H. Kim, D. Shin, and J. Kim, "A study on performance evaluation of fingerprint sensors," in *Audio and Video Based Biometric Person Authentication*, pp. 574–583, 2003.

[74] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *ACM Conference on Computer and Communications Security*, pp. 28–36, 1999.

[75] A. Juels and M. Wattenberg, "A fuzzy vault scheme," in *Proceedings of IEEE International Symposium on Information Theory, 2002.*, 2002.

[76] C. Soutar and G. J. Tomko, "Secure private key generation using a fingerprint," in *Proceedings of Cardtech/Securetech Conference*, vol. 1, pp. 245–252, May 1996.

[77] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. Kumar, "Biometric encryption^{TM} - enrollment and verification procedures," in *SPIE 98: In Proceedings of Optical Pattern Recognition IX*, vol. 3386, pp. 24–35, 1998.

[78] G. Davida, Y. Frankel, and B. Matt, "The relation of error correction and cryptography to an offine biometric based identication scheme," in *Proceedings of WCC99, Workshop on Coding and Cryptography, 1999.*, 1999.

[79] W. Zhang, Y.-J. Chang, and T. Chen, "Optimal thresholding for key generation based on biometrics.," in *ICIP '04: International Conference on Image Processing*, pp. 3451–3454, 2004.

[80] F. Monrose, M. K. Reiter, Q. Li, and S. Wetzel, "Cryptographic key generation from voice," in *SP '01: Proceedings of the 2001 IEEE Symposium on Security and Privacy*, (Washington, DC, USA), p. 202, IEEE Computer Society, 2001.

[81] A. Goh and D. C. Ngo, "Computation of cryptographic keys from face biometrics," in *Communications and Multimedia Security*, vol. 2828 of *LNCS*, pp. 1–13, 2003.

[82] D. C. Ngo, A. B. Teoh, and A. Goh, "Biometric hash: high-confidence face recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, pp. 771–775, June 2006.

[83] A. Ross, A. K. Jain, and J.-Z. Qian, "Information fusion in biometrics," in *Pattern Recognition Letters*, vol. 24, pp. 2115–2125, September 2003.

[84] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, (London, UK), pp. 89–105, Springer-Verlag, 1993.

[85] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, Maryland: Johns Hopkins University Press, 1983.

[86] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[87] V. N. Vapnik, *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

[88] C. Li, L. Khan, and B. Prabhakaran, "Real-time classification of variable length multi-attribute motions," *Knowledge Information Systems*, vol. 10, no. 2, pp. 163–183, 2006.

[89] X. min Tao, F. rong Liu, and T. xian Zhou, "A novel approach to intrusion detection based on SVD and SVM," *Industrial Electronics Society*, vol. 3, pp. 2028–2033, November 2004.

[90] Y. Wang, Y. Sun, M. Liu, P. Lv, and T. Wu, "Automatic inspection of small component on loaded PCB based on SVD and SVM," in *Mathematics of Data/Image Pattern Recognition, Compression, and Encryption with Applications IX.*, vol. 6315 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, September 2006.

[91] S. S. Kozat, R. Venkatesan, and M. K. Mihcak, "Robust perceptual image hashing via matrix invariants," in *International Conference on Image Processing*, pp. V: 3443–3446, 2004.

[92] S. Wang and Y. Wang, "Fingerprint enhancement in the singular point area," *IEEE Signal Processing Letters*, vol. 11, pp. 16–19, January 2004.

[93] J. H. Wegstein, "An automated fingerprint identification system," tech. rep., US Government Publication, 1982.

[94] D. Maio and D. Maltoni, "FVC2004: third fingerprint verification competition." `http://bias.csr.unibo.it/fvc2004/`, 2004.

[95] M. K. Mihçak and R. Venkatesan, "New iterative geometric methods for robust perceptual image hashing," in *DRM '01: Revised Papers from the ACM CCS-8 Workshop on Security and Privacy in Digital Rights Management*, (London, UK), pp. 13–21, Springer-Verlag, 2002.

[96] V. Licks and R. Jordan, "Geometric attacks on image watermarking systems," *IEEE MultiMedia*, vol. 12, no. 3, pp. 68–78, 2005.

[97] "K-Fold Cross Validation." `http://en.wikipedia.org/wiki/Cross-validation`.

[98] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[99] M. Li and I. K. Sethi, "SVM-based classifier design with controlled confidence," in *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1*, (Washington, DC, USA), pp. 164–167, IEEE Computer Society, 2004.

[100] R. Cappelli, "SFinGe: an approach to synthetic fingerprint generation," in *International Workshop on Biometric Technologies (BT2004)*, (Calgary, Canada), pp. 147–154, June 2004.

[101] H. Proença and L. A. Alexandre, "Toward non-cooperative iris recognition: A classification approach using multiple signatures," *IEEE Transactions on Pattern Analysis and Machine Intelligence, Special Issue on Biometrics*, vol. 9, pp. 607–612, July 2007. ISBN 0162-8828.

[102] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[103] "AT & T Databases of Faces." http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[104] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *IEEE Workshop on Applications of Computer Vision*, (Sarasota (Florida)), December 1994.

[105] F. Hao, R. Anderson, and J. Daugman, "Combining crypto with biometrics effectively," *IEEE Transactions on Computers*, vol. 55, no. 9, pp. 1081–1088, 2006.

[106] H. Proença and L. A. Alexandre, "UBIRIS: a noisy iris image database," in *ICIAP 2005: International Conference on Image Analysis and Processing*, vol. 1, pp. 970–977, 2005.

[107] C.-C. Li and K. S. Fu, "Machine-assisted pattern classification in medicine and biology," *Annual Review of Biophysics and Bioengineering*, vol. 9, pp. 393–436, 1980.

[108] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," in *IBM Systems Journal*, vol. 3, p. 40, IBM, 2001.

[109] M. Gupta, P. Judge, and M. Ammar, "A reputation system for peer-to-peer networks," in *NOSSDAV '03: Proceedings of the 13th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, (New York, NY, USA), pp. 144–152, ACM Press, 2003.

[110] S. Duri, M. Gruteser, X. Liu, P. Moskowitz, R. Perez, M. Singh, and J.-M. Tang, "Framework for security and privacy in automotive telematics," in *WMC '02: Proceedings of the 2nd international Workshop on Mobile Commerce*, (New York, NY, USA), pp. 25–32, ACM Press, 2002.

[111] E. F. Brickell, D. Chaum, I. B. Damgård, and J. van de Graaf, "Gradual and verifiable release of a secret," in *Proceedings of CRYPTO 87* (C. Pomerance, ed.), pp. 156–166, Springer-Verlag, 1988. Lecture Notes in Computer Science No. 293.

[112] G. Wang, "Generic non-repudiation protocols supporting transparent off-line TTP," *Journal of Computer Security*, vol. 14, no. 5, pp. 441–467, 2006.

[113] A. S. Patrick and S. Kenny, "From privacy legislation to interface design: Implementing information privacy in human-computer interfaces," in *Proceedings of Privacy Enhancing Technologies Workshop (PET2003), LNCS 2760*, 2003.

[114] J. Shao, Z. Cao, and L. Wang, "Efficient ID-based threshold signature schemes without pairings," *Cryptology ePrint Archive*, 2006. http://eprint.iacr.org/2006/308.pdf.

[115] C.-H. Wang and Y.-S. Kuo, "An efficient contract signing protocol using the aggregate signature scheme to protect signers' privacy and promote reliability," *SIGOPS: Special Interest Group on Operating Systems*, vol. 39, no. 4, pp. 66–79, 2005.

[116] B. Schneier and J. Riordan, "A certified e-mail protocol," in *ACSAC: 13th Annual Computer Security Applications Conference*, ACM Press, pp. 347–352, 1998.

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

[118] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of CRYPTO 84 on Advances in cryptology*, (New York, NY, USA), pp. 47–53, Springer-Verlag New York, Inc., 1985.

[119] U. Jendricke, M. Kreutzer, and A. Zugenmaier, "Mobile identity management," Tech. Rep. 178, Institut fr Informatik, Universitt Freiburg, October 2002.

[120] O. Kolsi and T. Virtanen, "Midp 2.0 security enhancements," in *HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 9*, (Washington, DC, USA), p. 90287.3, IEEE Computer Society, 2004.

[121] A. Wolfe, "Toolkit: Java is jumpin'," *Queue*, vol. 1, no. 10, pp. 16–19, 2004.

[122] Cellica Software Services, "Mobile database viewer (access,xls,oracle) for s60." `http://www.cellica.com/MobileDBViewer.htm`.

[123] A. Dix, T. Rodden, N. Davies, J. Trevor, A. Friday, and K. Palfreyman, "Exploiting space and location as a design framework for interactive mobile systems," *ACM Transactions on Computer Human Interaction*, vol. 7, no. 3, pp. 285–321, 2000.

[124] P. Coad and M. Mayfield, *Java Design: Building Better Apps and Applets*. Englewood Cliffs, NJ 07632, USA: Prentice-Hall, second ed., 1999.

[125] L. Loeb, *Secure electronic transactions: introduction and technical reference*. Norwood, MA, USA: Artech House, Inc., 1998.

[126] D. Wagner and B. Schneier, "Analysis of the ssl 3.0 protocol," in *WOEC'96: Proceedings of the Second USENIX Workshop on Electronic Commerce*, (Berkeley, CA, USA), pp. 4–4, USENIX Association, 1996.

[127] A. Bhargav-Spantzel, A. C. Squicciarini, M. Young, and E. Bertino, "Privacy requirements in identity management solutions," in *Twelfth International Conference on Human-Computer Interaction*, 2007.

[128] V. Samar, "Single sign-on using cookies for web applications," in *WETICE '99: Proceedings of the 8th Workshop on Enabling Technologies on Infrastructure for Collaborative Enterprises*, (Washington, DC, USA), pp. 158–163, IEEE Computer Society, 1999.

[129] "Web services trust language." `http://www.ibm.com/developerworks/library/specification/ws-trust/`.

[130] K. Lawrence and C. Kaler, "Web services security: SOAP message security," Tech. Rep. 1.1, OASIS Standard Specification, February 2006.

[131] "Internet2." `http://www.internet2.edu/`.

[132] ITU Telecommunication Standardization Sector, "The directory: Selected attribute types, ITU-T recommendation X.520," 1993.

[133] W. Duserick and F. Investments, "Whitepaper on liberty protocol and identity theft," in *Liberty Alliance Project*, 2004. `http://www.projectliberty.org/about/whitepapers.php`.

[134] M. Swift, J. Trostle, and J. Brezak, "Microsoft Windows 2000 kerberos change password and set password protocols." RFC 3244 (Informational), Feb. 2002.

[135] S. Brands, "Electronic cash systems based on the representation problem in groups of prime order," in *Preproceedings of Advances in Cryptology — CRYPTO '93*, pp. 26.1–26.15, 1993.

[136] IBM Zurich Research Laboratory, "Privacy-enhancing cryptography and pseudonym management." `http://www.zurich.ibm.com/security/privacy/`.

[137] Liberty Alliance Project, "Liberty protocols and schemas specification version 1.0." `www.projectliberty.org`, July 2002.

[138] E. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," in *CCS '04: Proceedings of the 11th ACM Conference on Computer and Communications Security*, (New York, NY, USA), pp. 132–145, ACM Press, 2004.

[139] S. L. Kinney, *Trusted Platform Module Basics: Using TPM in Embedded Systems (Embedded Technology)*. Newnes, 2006.

[140] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *Lecture Notes in Computer Science*, vol. 2139, pp. 213–240, 2001.

[141] K. G. Paterson, "Identity-based signatures from pairings on elliptic curves," *Cryptology ePrint Archive, Report*, 2002.

[142] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," *SIAM Journal of Computing*, vol. 32, no. 3, pp. 586–615, 2003.

[143] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf, "Pseudonym systems," in *Selected Areas in Cryptography 1999*, pp. 184–199, 1999.

[144] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. Kumar, "Biometric encryption^{TM} using image processing," in *SPIE 98: In Proceedings of Optical Security and Counterfeit Deterrence Techniques II*, vol. 3314, pp. 178–188, 1998.

[145] J. Daugman, "Biometric personal identification system based on iris analysis," in *United States Patent*, 1994.

[146] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[147] D. Bleichenbacher and P. Q. Nguyen, "Noisy polynomial interpolation and noisy Chinese remaindering," *Lecture Notes in Computer Science*, vol. 1807, pp. 53–77, 2000.

[148] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure smartcard based fingerprint authentication," in *WBMA '03: Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications*, (New York, NY, USA), pp. 45–52, ACM Press, 2003.

[149] S. Yang and I. Verbauwhede, "Automatic secure fingerprint verification system based on fuzzy vault scheme," in *ICASSP '05: Proceedings of the Acoustics, Speech, and Signal Processing*, vol. 5, (Philadelphia, USA), pp. 609–612, March 2005.

[150] Y. C. Feng and P. C. Yuen, "Protecting face biometric data on smartcard with reed-solomon code," in *Proceedings of CVPR Workshop on Privacy Research In Vision*, (New York, USA), p. 29, June 2006.

[151] Y. Chung, D. Moon, S. Lee, S. Jung, T. Kim, and D. Ahn, "Automatic alignment of fingerprint features for fuzzy fingerprint vault," in *In Proceedings of Conference on Information Security and Cryptology*, (Beijing, China), pp. 358–369, Dec. 2005.

[152] M. Freire-Santos, J. Fierrez-Aguilar, and J. Ortega-Garcia, "Cryptographic key generation using handwritten signature," in *Proceedings of SPIE: Biometric Technology for Human Identification III* (P. J. Flynn and S. Pankanti, eds.), vol. 6202, 2006.

[153] U. Uludag and A. Jain, "Securing fingerprint template: Fuzzy vault with helper data," in *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, (Washington, DC, USA), p. 163, IEEE Computer Society, 2006.

[154] K. Nandakumar, A. K. Jain, and S. Pankanti, "Fingerprint-based fuzzy vault: Implementation and performance," in *IEEE Transactions on Information Forensics and Security, 2007 (To appear)*, 2007.

[155] F. Monrose, M. K. Reiter, and S. Wetzel, "Password hardening based on keystroke dynamics," in *CCS '99: Proceedings of the 6th ACM conference on Computer and communications security*, (New York, NY, USA), pp. 73–82, ACM Press, 1999.

[156] A. Lumini and L. Nanni, "An improved biohashing for human authentication," *Pattern Recognition*, vol. 40, no. 3, pp. 1057–1065, 2007.

[157] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You, "An analysis of biohashing and its variants," *Pattern Recognition*, vol. 39, no. 7, pp. 1359–1368, 2006.

[158] U. Uludag, S. Pankanti, S. Prabhakar, and A. Jain, "Biometric cryptosystems: Issues and challenges," in *Proceedings of the IEEE, Special Issue on Enabling Security Technologies for Digital Rights Management,,2004.*, vol. 92, 2004.

[159] "Gnutella." `http://www.gnutella.com`.

[160] "Attentiontrust.org." `http://www.attentiontrust.org`.

[161] B. Cox, J. D. Tygar, and M. Sirbu, "NetBill security and Transaction Protocol," in *USENIX Workshop on Electronic Commerce*, pp. 77–88, July 1995.

[162] G. J. Simmons and G. B. Purdy", "Zero-knowledge proofs of identity and veracity of transaction receipts," *EUROCRYPT '88: Workshop on the Theory and Application of Cryptographic Techniques*, vol. 330, p. 35, May 1988.

[163] T. Coffey and P. Saidha, "Non-repudiation with mandatory proof of receipt," *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 1, pp. 6–17, 1996.

[164] S. F. Mjolsnes and C. Rong, "Localized credentials for server assisted mobile wallet," *ICCNMC'01: International Conference on Computer Networks and Mobile Computing*, vol. 00, p. 203, 2001.

[165] J.-P. Boly, A. Bosselaers, R. Cramer, R. Michelsen, S. F. Mjolsnes, F. Muller, T. P. Pedersen, B. Pfitzmann, P. de Rooij, B. Schoenmakers, M. Schunter, L. Vallee, and M. Waidner, "The ESPRIT project CAFE - high security digital payment systems," in *ESORICS*, pp. 217–230, 1994.

[166] S. Mizuno, K. Yamada, and K. Takahashi, "Authentication using multiple communication channels," in *DIM '05: Proceedings of the 2005 workshop on Digital identity management*, (New York, NY, USA), pp. 54–62, ACM Press, 2005.

[167] K. Rannenberg, "Identity management in mobile cellular networks and related applications," Information Security Technical Report 9, Johann Wolfgang Goethe University Frankfurt, January 2004.

[168] D. Hennessy, "The value of the mobile wallet." White Paper, December 2003. `www.valista.com`.

[169] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "SETI@home: An experiment in public-resource computing," *Communications of the ACM*, vol. 45, pp. 56–61, November 2002.

[170] "The great internet mersenne prime search." `http://www.mersenne.org/`.

[171] P. Golle and I. Mironov, "Uncheatable distributed computations," *Lecture Notes in Computer Science*, vol. 2020, pp. 425+, 2001.

[172] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai, "Zero-knowledge from secure multiparty computation," in *STOC '07: Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, (New York, NY, USA), pp. 21–30, ACM Press, 2007.

[173] Y. Desmedt, G. Di Crescenzo, and M. Burmester, "Multiplicative non-abelian sharing schemes and their application to threshold cryptography," in *Advances in Cryptology — Asiacrypt '94, Proceedings (Lecture Notes in Computer Science 917)* (J. Pieprzyk and R. Safavi-Naini, eds.), pp. 21–32, Springer-Verlag, 1995.

[174] Y. G. Desmedt and Y. Frankel, "Perfect homomorphic zero-knowledge threshold schemes over any finite abelian group," *SIAM Journal of Discrete Mathematics*, vol. 7, no. 4, pp. 667–679, 1994.

[175] A. Juels, D. Molnar, and D. Wagner, "Security and privacy issues in e-passports," in *SecureComm 2005, Security and Privacy for Emerging Areas in Communications Networks, 2005.* (IEEE, ed.), pp. 74–88, September 2005.

[176] S. S. Chan, R. J. Wolfe, and X. Fang, "Issues and strategies for integrating HCI in masters level MIS and e-commerce programs," *International Journal of Human-Computer Studies*, vol. 59, no. 4, pp. 497–520, 2003.

[177] Federal Trade Commission, "Laws: Identity theft." `http://www.ftc.gov/bcp/edu/microsites/idtheft/law-enforcement/laws.html`.

[178] M. Frank, "Identity theft prevention and survival." `http://www.identitytheft.org/`.

[179] "Gridsphere portal framework." `www.gridsphere.org`.

APPENDICES

APPENDICES

Appendix A: State and Federal Laws Designed to Protect Personal Information

Increased federal and state legislation regarding identity theft has brought a heightened awareness to identity theft in general and the special status of an individual's SSN as an identifier in particular. For instance, the Identity Theft and Assumption Deterrence Act of 1998 makes identity theft a federal crime (18 U.S.C. § 1028 (2003)). The purpose of this statute is to criminalize the act of identity theft itself, before other crimes are committed. Under this law, identity theft occurs when a person "knowingly transfers, possesses or uses, without lawful authority, a means of identification of another person with the intent to commit, or to aid or abet, or in connection with, any unlawful activity that constitutes a violation of federal law, or that constitutes a felony under any applicable state or local law." 18 U.S.C. § 1028(a)(3)(7). Under this law, a name or SSN is considered a "means of identification." (18 U.S.C. § 1028(c)(3)(C)(3)(A)). States have attempted to be proactive with the crime of identity theft as well. In Indiana, for example, a person who "knowingly or intentionally obtains, possesses, transfers, or uses the identifying information of another person" without consent and has an intent to harm or defraud another person or assume the other person's identity commits identity deception (Ind. Code § 35-43-5-3.5 (2004)). Under Indiana's law, "identifying information" specifically includes a SSN (Ind. Code § 35-43-5-1(i)). Growing recognition of the availability of SSN and that number's ubiquitous use as a means of identifying a person for a number of purposes has spurred state legislation trying to combat the careless and cavalier use of the number. Many of the new laws enacted at the state level contain provisions addressing the circumstances under which SSN and other personally identifying information can be disclosed to third parties, confidential destruction of papers and electronic media containing SSN and personally identifying information of cus-

tomers, and requirements for encryption of SSN and other sensitive personally identifying information held in electronically stored mediums [177, 178].

Appendix B: VeryIDX Web-Based Implementation Prototype

In this appendix we provide the architectural design and prototype implementation details of VeryIDX registrar and other key components. We also perform a performance and storage analysis of the prototype system.

B.1 System Architecture

To implement the VeryIDX system we developed components for three main entities of this system, namely *registrar, SP* and *principal*. Several main considerations were taken into account in the design of VeryIDX.

1. The requirement to *minimally extend* the existing components used for e-commerce transactions.

   First, as principals and SPs should have easy access to the registrar we made our system web-based. Thus, no client side software installation is needed. Second, requiring modification to the current verification processes of SPs would not be desirable because of backward compatibility and scalability issues. Therefore we provide add-on modules for SP to join the VeryIDX system. Furthermore our system does not affect the legacy interactions between SPs and principal's.

2. Providing *de facto interoperation*. VeryIDX achieves interoperation using a few registrar components. Different SP can specify their requirements according to their service policies and subsequently use the registrar to obtain relevant and reliable information when they have to make decisions for identity verification and trust establishment.

3. Providing *scalable and interchangeable building blocks*. A modular application is composed of smaller, separated modules that are well isolated. Thus, it makes easier

to develop and manage than tightly coupled application. We adopt modularization so it is easy to update component and simple to add new functionality.

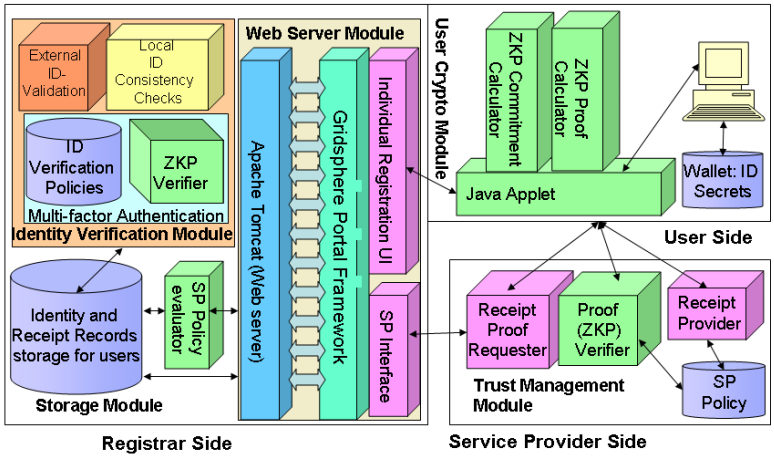


Fig. B.1. System architecture of web-based VeryIDX.

Figure B.1 shows the general architecture of the VeryIDX system. In the following we describe system architecture, implementation details and subcomponents of the three entities, namely registrar, SP and principal.

Registrar Side

Registrar handles principals' request to add or extend receipts and other identity attributes. It manages principals' identity record (IdR) and receipt record (RREC). Registrar comprises four key modules that are described as follows.

1. *Webserver module.* This module comprises servlet container and the implementation of registrar. The servlet container accepts principals' connection and relays it to the registrar components. It is in charge of processing the principals' request, e.g, showing dynamic receipts, allowing principals to add new receipt and so on. We used

Gridsphere Portal Framework [179] to create dynamic web pages because of its extensive built-in servlets, user management portlet capabilities and reliability. As a servlet container, Jakarta Tomcat has been selected for its robustness.

2. *Record storage.* To store the principals records, namely IdR and RREC, we use Oracle 10g database. Identity related information such as registrar's public key and some public parameters for example, p, q, g and h required for ZKPK protocols are also stored in the database.

3. *Identity verification module.* The identity verification module checks the correctness of the claimed identity. It performs aggregate ZKPK for multi-factor identity verification, before this principal is allowed access to its IdR or RREC. It is also used to assess the assurance level on a given attribute. For example, in the case of history based attributes, this module checks whether a principal logged in at the registrar is the same as the one indicated in the receipt obtained from SP using the receipt assurance steps specified in Protocol 1 of Chapter 5. Secure Socket Layer (SSL) is deployed for the secure traffic using the Java Secure Socket Extension (JSSE) package.

4. *SP policy evaluator.* This module receives as input the verification and trust establishment policies of the SP. Given the IdR and RREC of the principal it then identifies the potential attributes and receipts of the principal that can satisfy the policy requirements. Thereafter the result is presented to the principal, so it can construct the proofs based on the selected tuples of the IdR and the RREC.

Principal Side

The key module used at the principals end is the *User Crypto Module*. This module consists of two components namely *ZKP commitment Calculator* and *ZKP Proof Calculator*. The ZKP commitment calculator computes commitment of any given attribute related to either identity or receipt information. A critical requirement is that the secrets involved in creating the commitment should not leak outside the principals' machine. Only the final commitment is revealed once the computation is complete and the principals secrets are

stored at the principals local machine. The ZKP Proof Calculator creates a proof object that can be submitted to the SP or registrar. Using this object, the verifier can carry out the ZKPK proof verification.

Java Applets [124] are used to implement both components. Java applets can communicate not only with the servlet in the registrar but also with the principals' local file system. In addition, some parameters about principals can be passed to the applets from JSP. Applets have some access limitations to the user file system because they are not part of the local system. In our system, we use signed applets that can be allowed to access local filesystem. In addition, principals are required to put a policy file to enable an applet to access local file having principals' secret. We devised a logical structure named 'wallet' where the principal securely stores its personal information and secrets. When principals create commitments, the secret information is stored at the wallet to be used later in ZKPK. The secret information in the wallet is never revealed to the registrar nor the SP. We use Java Serialization technique[1] for applet to servlet communication. This approach does not require the system to deploy additional protocols for the data transmission. The receiver simply needs to get serialized stream and recover the same objects the sender transmitted.

Service Provider Side

The *Trust Management Module* is the main extension required to the SP. Such module has four main components. The first is the SP *policy database*; such database is accessed by the *proof requester* component. This component is responsible for creating the conditional statements required to verify identity and establish trust. Once the proof is provided, the *proof verifier* takes the proof object that may consist of clear attribute values or cryptographic proofs. The proofs are verified to get a boolean value determining the verification or trust establishment decisions. The last component is the *receipt provider* that issues the receipts when the e-commerce transaction is completed (See Chapter 5).

[1]Serialization saves the current state of objects to a stream and restores an equivalent object from the stream. Stream can hold data in a persistent container (disk) or transient container (RAM).

B.2 Implementation Analysis

Our system is web-based therefore the response time mainly depends from the network speed. The message size related to the cryptographic computations is analyzed in Section 5.4.3. The computation time and the storage requirements of our protocols are minimal as detailed below. We have carried out our experiments on an Intel Core duo 2GHz and 2G RAM, and server had 2.8GHz Pentium D CPU with 1G RAM.

Average time to execute AgZKPK using applets. From our experiments, the average time to log onto the registrar, using user name and password, over 100 iterations takes less than 1 sec. Likewise, the time to download an applet takes around one second under a network whose average data receive rate is 928 Bytes per second (relatively slow connection). To extend a receipt, the applet running on client receives a tag and a value pair from the registrar that are then used to calculate the commitment. Excluding the principals interaction time, to calculate a single commitment takes on an average 0.011 sec.

Summing up the total time including commitment computation, transmission to the server and receipt of the reply, the average time to extend a receipt, takes 1.03 seconds. The average time for the applet to create proofs is .020 seconds as illustrated in Figure 5.4 and detailed in Section 5.4.3 of Chapter 5.

At the registrar side, one of the major functions is to store principals record into the database. Every time a principals commitment arrives to the server, the registrar makes a connection to the Oracle 10g database by issuing one INSERT statement. The average insertion time was measured 0.5 sec. Finally, as illustrated in Figure 5.5 of Chapter 5 the proof verification at the registrar, for 50 aggregated identifier proofs takes .103 seconds at the registrar.

Average storage needed at the principal and the registrar. Our implementation requires less than 6M bytes of disk space for the portal codes under the tomcat directory at server side. At the client side, principals' secret needed for the commitments are recorded at the VeryIDX.wallet and its size increases around 5KB for each commitment. The registrar stores principals' record into Oracle database. For the other registrar components, the

minimum space required is about 50MB for tomcat excluding disk space for the Operating System. The RREC of a principal is on an average 67M bytes for 106 receipts with one cryptographic commitment. Each commitment value takes 31 digit characters on an average.

B.3 Illustrative Example of the VeryIDX Receipt Based System

The main steps related to the cryptographic computations related to aggregate ZKPK using IdR or RREC are similar. Therefore in the following we provide an example scenario of how a principal would use suach a system to manage history based attributes encoded as receipts from online transactions as presented in Chapter 5.

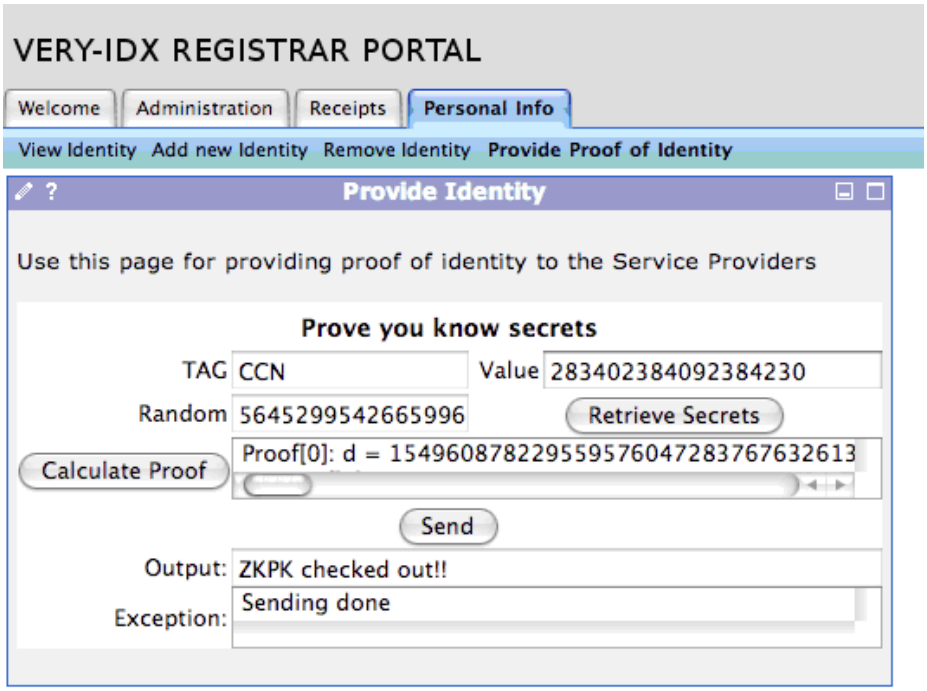


Fig. B.2. Applet for creating ZKPK for identity attributes.

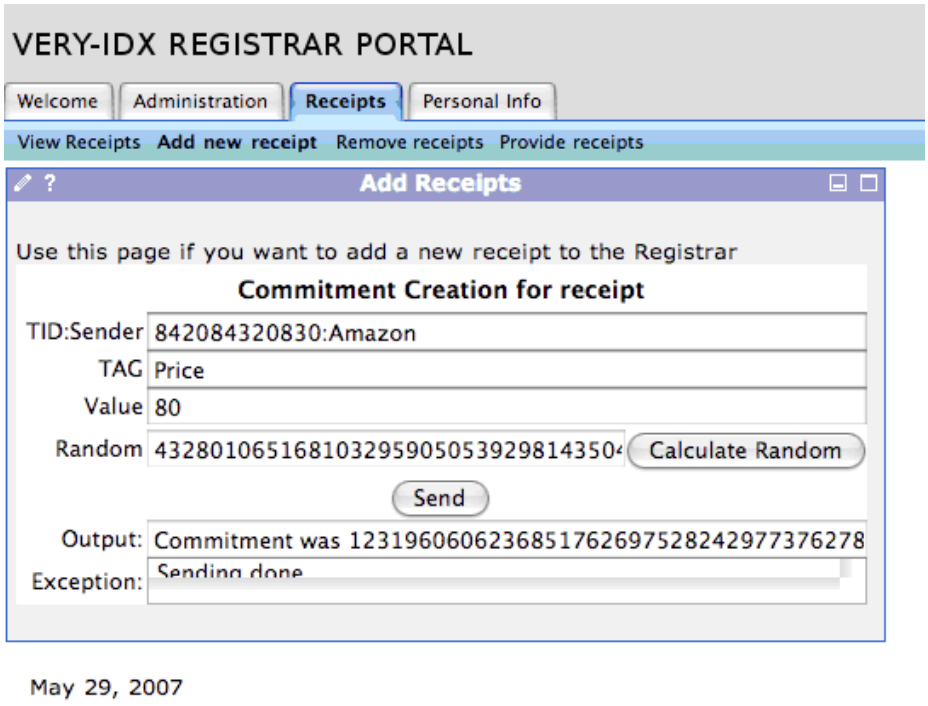


Fig. B.3. Applet for creating commitments for x-receipts.

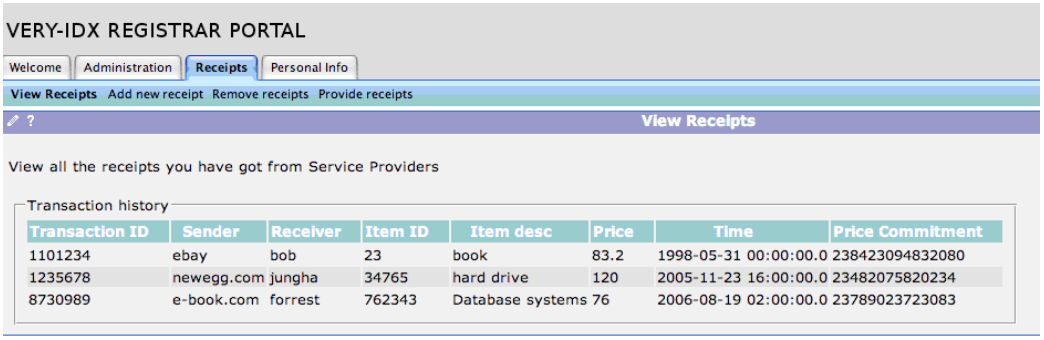


Fig. B.4. Registrar portal view of receipts in RREC.

Consider a scenario when a principal $Alice$ has bought a book from VeryIDX enabled $eFollets$ and now wants to opt-in to add the receipt of this transaction to her RREC at the registrar. She uses Protocol 1 of Chapter 5 (See also Figure 5.1). Once she has sent her intention to get the receipt to SP (Step 1) then she logs on to the registrar using her SSO ID and password. Her registrar requires multi-factor identity verification (Step 2) so it requires $Alice$ to prove she knows the secrets corresponding to her credit card number commitment stored in her IdR. She runs the proof calculator applet (See Figure B.2) where she can automatically retrieve the required secrets by clicking the "Retrieve Secrets" button. Once the secrets are retrieved she clicks on "Calculate Proof" to calculate the proof object. Finally she sends the proof object to the registrar that is used to verify its the correctness of the proof. If the proof is verified correctly, the reply of the registrar appears on the principals' applet.

As a next step the registrar generates the ρ_{submit} used eventually by the SP to give $Alice$ the correct receipt. Note here that SP creates a TRANSACTION ID that is unique to this SP. Subsequently $Alice$ can add this receipt using Step 5 of Protocol 1. At any point $Alice$ can view her RREC at registrar by logging on to her registrar using step 2 of Protocol 1 (See Figure B.4).

Once the e-receipt is submitted, $Alice$ extends this receipt using Protocol 3. More specifically she creates a cryptographic commitment corresponding to PRICE attribute of her receipt. To do this, she logs on to her account at the registrar and this time she runs the commitment creation applet (See Figure B.3). Here the main requirement is that the principal should have unique tag values corresponding to each commitment. The TAG is the combination of the TRANSACTION ID, SENDER and the type of attribute being committed (on this case the price). The random needed at Step 2 of Protocol 3 is computed when she clicks the "calculate Random" button. She can then send this commitment to the server. The commitment can be subsequently used to create proofs as illustrated in Figure B.2.

VITA

VITA

Abhilasha Bhargav-Spantzel was born in Calcutta, India. She completed her schooling from Loreto Convent, Lucknow, India. She pursued her undergraduate studies at Purdue University. Abhilasha obtained her bachelor's degree in computer science and mathematics with minors in physics and psychology in December 2002.

In August 2003, she enrolled in the Department of Computer Science of Purdue University. She was a resident fellow intern at Symantec in the summer of 2004. Since 2005, Abhilasha has been involved with the Liberty Alliance identity theft special interest group. She interned at the IBM Zurich research lab for three months in the summer of 2006. Following that, she interned at Intel for three months in fall of 2006. Since 2007, Abhilasha has been involved with the Institute for Information Infrastructure Protection. She received the degree of Doctor of Philosophy in December 2007 with Elisa Bertino as her major professor. Her research interests are computer security, applied cryptography, biometrics, electronic commerce, with a focus on privacy and protection of identity.