The Design and Applications of a Privacy-Preserving Identity and Trust-Management System

by

Mohammed Hussain

A thesis submitted to the
School of Computing
in conformity with the requirements for
the degree of Doctor of Philosophy

Queen's University
Kingston, Ontario, Canada
April 2010

Your file Votre référence
ISBN: 978-0-494-65112-4
Our file Notre référence
ISBN: 978-0-494-65112-4

Abstract

Identities are present in the interactions between individuals and organizations. Online shopping requires credit card information, while e-government services require social security or passport numbers. The involvement of identities, however, makes them susceptible to theft and misuse. The most prominent approach for maintaining the privacy of individuals is the enforcement of privacy policies that regulate the flow and use of identity information.

This approach suffers two drawbacks that severely limit its effectiveness. First, recent research in data-mining facilitates the fusion of partial identities into complete identities. That holds true even if the attributes examined are not, normally considered, to be identifying. Second, policies are prone to human error, allowing for identity information to be released accidentally.

This thesis presents a system that enables an individual to interact with organizations, without allowing these organizations to link the interactions of that individual together. The system does not release individuals' identities to organizations. Instead, certified artificial identities are used to guarantee that individuals possess the required attributes to successfully participate in the interactions. The system limits the fusion of partial identities and minimizes the effects of human error. The concept of using certified artificial identities has been extensively researched. The system,

however, tackles several unaddressed scenarios.

The system works not only for interactions that involve an individual and an organization, but also for interactions that involve a set of individuals connected by structured relations. The individuals should prove the existence of relations among them to organizations, yet organizations cannot profile the actions of these individuals. Further, the system allows organizations to be anonymous, while proving their attributes to individuals. Reputation-based trust is incorporated to help individuals make informed decisions whether to deal with a particular organization.

The system is used to design applications in e-commerce, access control, reputation management, and cloud computing. The thesis describes the applications in detail.

Co-Authors

The paper "M. Hussain and D. B. Skillicorn. Persona-based identity management: A novel approach to privacy protection, *in Proceedings of the 13th Nordic Workshop on Secure IT Systems* (2008), Technical University of Denmark, Copenhagen, Denmark, pp. 201–212" is based on Chapter 3 and 4.

The technical report "D. B. Skillicorn and M. Hussain. Personas: Beyond Identity Protection by Information Control, Technical report, *commissioned by the Office of the Privacy Commissioner of Canada* (2009). School of Computing, Queen's University, Kingston, Canada, Retrieved Jan 2010, from `research.cs.queensu.ca/home/skill/opccreport.pdf`" is based on Chapter 3 and 4.

The paper "M. Hussain, D.B. Skillicorn. Guarantee-Based Access Control, *in Proceedings of the IEEE International Conference on Computational Science and Engineering* (2009), IEEE Computer Society, vol. 3, pp. 201–206" is based on Chapter 6.

The book chapter "M. Hussain and D. B. Skillicorn, Securing mobile agent systems through collaboration, *in J.M. Seigneur and A. Slagell (Ed.)*, *Collaborative Computer Security and Trust Management* (2009), IGI, pp. 154–180" is part of the background.

Dedication

In memory of my Grandmother Fatima.

Acknowledgements

I offer my thanks first and foremost to GOD for paving the way for this task.

I thank my supervisor Prof. David B. Skillicorn for his excellent supervision, extensive help, and great advice. I thank my supervisory committee for their assistance and precious feedback. Thanks goes to my defense committee for their valuable time and effort.

I thank the School of Computing for the wonderful experience I had during my M.Sc. and Ph.D. programs. Thanks to my professors and colleagues. Special thanks to Bader Al-Manthari and Hanady Abdulsalam for their kind assistance.

I finally thank my parents, sisters and brothers. Their dedicated support and love kept me going.

Statement of Originality

I, Mohammed Hussain, certify that the work presented in this thesis is original unless otherwise noted. Any published (or unpublished) ideas and/or techniques from the work of others are fully acknowledged in accordance with the standard referencing practices.

Contents

List of Figures

List of Acronyms

ABAC	Attribute-Based Access Control
ACL	Access-Control List
ARM	Anonymous-Reputation Management
B2B	Business to Consumer
B2C	Business to Business
CCA2	Chosen Cyphertext Attacks
DA	De-anonymization Authority
DLDH	Decisional Linear Diffie Hellman
FIMS	Federated Identity-Management System
GBAC	Guarantee-Based Access Control
HIDS	Hidden IDentity-based Signatures
IBAC	Identity-Based Access Control
IDP	Identity Provider
IMS	Identity-Management System
P2P	Peer-to-Peer
PP	Persona Provider

RBAC	Role-Based Access Control
RMC	Reputation-Management Component
SAML	Security Assertion Markup Language
SDH	Strong Diffie Hellman
SP	Service Provider

Chapter 1

Introduction

1.1 Motivation

Identities may be represented by names, but are usually more complex. An identity may include emails and social network accounts, driver licenses, passports, social securities, and taxpayer numbers. These pieces of information are considered identities, since they can uniquely identify individuals. For example, passports are unique per person, in a given country. Moreover, information that is usually considered non-identifying, like gender and education, can become identity too. The combination of few partial identities, like postal code and age, may identify an individual in a given context.

Identities are an important aspect of our life. We use them to associate ourselves with ethnicities, religions, and ideologies. Identities are also used when people interact with each other, and with organizations, such as shops and governments. Organizations need identity information of individuals to authenticate them and regulate access to resources.

Interactions among individuals and organizations are inclined to involve identity information, which creates considerable economic and privacy risks when this information is lost or misused. The cost of identity theft, during 2008, is estimated to be $45 billion in the US [2] and £1.2 billion in the UK [46]. A published survey [1] states that more than 60 percent of Americans are extremely worried about their privacy when shopping online. This represents a significant increase from the 47 percent recorded in 2006. Loss and misuse of identities also lead to profiling individuals. When an individual's partial identities are linked together, many of the once-private actions are traced back to that individual. Organizations at which loss or misuse occur are affected as well, since individuals lose trust in these organizations. Protecting individuals' privacy is, therefore, essential for both individuals and organizations.

Research on privacy has useful applications in other areas as well. For example, privacy has a close relation to trust management, where an enhancement in one leads to an enhancement in the other. On the one hand, reputation-based trust techniques depend on ratings. In the context of individuals rating organizations, privacy plays a key role in encouraging individuals to rate. On the other hand, trust management allows individuals to determine the level of trust they should place in an organization. This helps individuals avoid interacting with untrusted organizations; and thus, individuals deprive these organizations from the needed identity information to profile individuals.

1.2 Problem Statement

Privacy is normally conceived as a problem of regulating the flow of identity information. Safe flows, those which do not contain identifying information, are allowed to

occur. Flows that do contain identifying information are filtered. Several drawbacks make this assumption and strategies based on it inadequate to protect individual privacy.

Recent advances in data-mining and fusion technologies enable partial identities to be associated with one another, and thus, good approximations to complete identities to be computed. What is alarming is the ability to achieve this even if each partial identity does not contain data that would normally be considered as identifying. For example, the work of Frankowski *et al.* [34] and Narayanan *et al.* [64] shows that anonymized users' records in movie-rating datasets can be traced back to the actual users with the help of a very small amount of auxiliary information (for example, a user discussing a movie in a public forum). The genomic data of anonymized patients have been linked back to the actual patients [59].

Customers may deal with seemingly different organizations and businesses that are actually part of the same conglomerate, and hence feel entitled to share information about their customers. Customers may provide only a little personal information to each organization, but it may be enough to build a complete profile.

Policies are prone to human error, allowing for identity information to be released accidentally. Once this has been done, there is no way to call it back. It only takes one ill-designed policy to undermine the privacy of a whole system. The e-commerce industry is full of examples where the information from millions of credit cards was disclosed due to inappropriate decisions [3, 77].

The above findings allow us to state two ways that data-mining techniques can be used to attack individuals' privacy: associate records with each other; and de-anonymize records with the help of auxiliary information.

The Linkability Problem. Given a dataset of anonymous records, which belongs to a set of individuals, it is possible to associate the records of an individual with each other; that is, computing profiles of individuals.

The De-Anonymization Problem. Given a dataset of associated anonymous records, where the records of each individual are associated with each other, it is possible to re-identify individuals in other datasets, with the help of auxiliary information from the other datasets.

The two problems have motivated significant research in the area of privacy and anonymity [7, 22, 41, 57, 78, 82]. The main theme of the research in this area is allowing individuals to interact anonymously with organizations, while preventing organizations from profiling individuals. However, there are several topics that remain unaddressed.

Privacy is mostly described in the scenario consisting of an individual interacting with an organization. Current work on privacy does not allow a set of individuals, connected by a set of relationships, to interact anonymously with an organization. This is used to require that a specified set of individuals in a specified relation must all participate for the interaction to be successful. For example, using a business account can require the simultaneous action of several specified company officers. Currently, a group of individuals may prove the existence of relations among group members to organizations, anonymously. However, if the same group revisits the same organization, the organization will be able to link the two different visits to the same group. This allows organizations to link the actions of a group together. Frankowski *et al.* [34], Malin *et al.* [59] and Narayanan *et al.* [64], show that linkable actions are used to build profiles and may lead to de-anonymization.

Another topic tackled by this thesis is the need of some organizations to be anonymous while interacting with individuals. In current Business-to-Consumer (B2C) settings, businesses may interact anonymously with consumers, but with the help of arbitrageurs. An example of this setting is the "name your price" feature by businesses, such as Priceline, where consumers are allowed to bid for hotel rooms. These arbitrageurs act as intermediaries who guarantee for consumers that the offers by the anonymous businesses are trustworthy. This makes the arbitrageurs not only providers of a market place, but also guarantors of businesses' offers. The disadvantages of such a setting are twofold: arbitrageurs are entitled to charge the businesses; and consumers must trust the arbitrageurs, since consumers cannot verify the offers. Allowing businesses and consumers to interact anonymously, without reliance on arbitrageurs, makes B2C interactions more profitable for businesses and more encouraging for consumers.

Incorporating trust management is another topic that this thesis handles. Trust and privacy are, usually, addressed independently. Addressing privacy and trust in a single framework is more efficient, since the two subjects are related. In the Semantic Web [11] and the Semantic Social Web, trust is an integral part for automatic service discovery and invocation. Therefore, such a framework has a profound application in the Semantic-Web setting.

Finally, our work tackles some issues in reputation management. Reputation-based trust is based on individuals explicitly rating products and services. Several problems exist with that approach. Individuals may not feel comfortable recommending certain products. For example, individuals may not feel safe recommending products related to religion and politics. If individuals neglect rating an organization,

that organization gains no reputation. By convincing individuals to use a service, an organization should gain some reputation (the fact that individuals had some trust in that organization in the first place). The next subsection states our thesis and contributions.

1.3 Thesis and Contributions

To overcome the problems of fusing identities and profiling individuals, we suggest the use of *artificial identities*, henceforth called personas. Personas represent individuals in transactions, but without carrying any identity information. Instead, a persona is just a way to assure organizations that there is a guarantor for the individual possessing that persona. Personas are therefore much more difficult to fuse.

We can think of a persona as acting as an identity representing someone in one or more interactions, for example, an email address, or a credit card. Personas also allow for a set of individuals to interact anonymously with an organization, while proving the relation among these individuals.

Personas use public-key cryptography; specifically, personas utilize the hidden identity-based signature scheme developed by Kiayias *et al.* [52]. The scheme is used to provide the needed cryptographic constructs to implement personas.

The main objective of this dissertation is to reduce the ability of data-mining techniques to fuse identity information, without affecting accountability. In other words, the objective is protecting the privacy of individuals, while maintaining the security of organizations. The examples by Frankowski *et al.* [34], Malin *et al.* [59] and Narayanan *et al.* [64] show the de-anonymization of individuals in movie-rating and patient datasets. Although protecting anonymity in these datasets is not part

of this thesis, we use these examples since the same techniques can be used to de-anonymize individuals in general.

Some techniques use artificial identities to enhance the privacy of individuals; however, we take a new approach and tackle new issues, as stated in the problem statement. The following are the contributions of this thesis.

1. Extending the notion of privacy to include the scenario where a group of individuals in a structured relationship interact with an organization. Current research on privacy focuses on protecting the privacy of an individual interacting with an organization. This contribution prevents the de-anonymization of individuals, even if these individuals interact with organizations as groups.

2. Allowing businesses to sell their products anonymously to consumers, while enabling consumers to verify the offers without the need to have trust in arbitrageurs. This helps businesses to avoid being charged by arbitrageurs to act as guarantors. This contribution prevents the de-anonymization of individuals, as well as businesses.

3. Presenting a technique to enable individuals to rate products and services in a private manner. The technique allows us to incorporate the notion of trust, specifically, reputation management. While privacy and trust are usually addressed separately, this contribution helps addressing the two subjects in one framework.

We present a system that achieves the contributions stated above. The system is based on the cryptographic scheme of Kiayias *et al.* [52]. The scheme enables

individuals to participate in secure unlinkable interactions with organizations. The scheme operations are modified to implement the contributions.

1.4 Thesis Organization

Chapter 2 describes the background information and related work to this research. The chapter provide the needed definitions and terminology to understand the rest of the thesis. It surveys the different approaches to identity management and provides detail description of several systems from each approach. The background and related work relevant to a specific chapter are discussed in the corresponding chapter.

Chapter 3 presents our secure anonymous interaction protocol. The chapter provides an abstract view of the requirements, architecture, operations, and building blocks. Sample scenarios are used to illustrate the course of action among the various entities.

Chapter 4 provides a more concrete view of our work. The chapter presents the required cryptographic support to implement the required operations and building blocks. This also includes the mapping between the abstract and concrete models.

Chapter 5 applies personas in e-commerce. The chapter shows how privacy and anonymity are enhanced by personas. This is illustrated using B2C and B2B settings.

Chapter 6 applies personas in access control. The chapter discusses the design and applications of basing access control decisions on the relations among individuals.

Chapter 7 applies personas in reputation-based trust. The chapter builds an anonymous reputation-management system.

Chapter 8 applies personas in a cloud-computing environment. A cloud-based identity-management system is constructed. The system utilizes the computational

power of cloud-computing to enhance the privacy of individuals.

Finally, Chapter 9 summarizes the contributions of this thesis. It highlights the limitations and possible enhancements of this work; then concludes the thesis.

Chapter 2

Background and Related Work

This chapter begins with defining the concept of *identity* in Section 2.1. In Section 2.2, current identity-management solutions are categorized and described in detail. Section 2.3 focuses on solutions that are related to our work. The limitations of the related work are presented in Section 2.4.

2.1 Identities

The definition of identity depends on the discipline and the required level of depth. In psychology and other disciplines, the definition is more theoretical and generic than in computer science and engineering. We use a simple and more practical definition of identity [47]: the set of attribute values that uniquely label a single individual in some specific context. Context refers to the circumstances and environment at which identity is defined. For example, in a university (context), a student can be identified by a student name and a student number (attributes). It is worth mentioning that the set of attributes varies from an individual to another.

Context determines the importance of an attribute in identifying individuals. For example, first names do not constitute unique identifiers for individuals within a country. However, within a midsize company, first names are much more identifying.

The attributes that are used to form an identity fall into three categories:

1. Attributes associated with physical and mental existence of an individual. This category includes fingerprints, iris patterns (physical), as well as skills that are hard to learn such as fluency in a particular language, and writing poetry (mental).

2. Attributes certified by a trusted party. Example attributes certified by the government are the name (authenticated by reference to a birth certificate) and citizenship. Such attributes often carry with them a set of *rights*.

3. Attributes chosen by the individual to act as (part of) their identity, for example a nickname or screen name.

These attributes can discriminate between individuals in a given context – knowing the value of even a single attribute may be enough to identify that individual in that context. When individuals participate in online shopping, they must provide values for attributes such as credit card numbers. Applying for a job requires showing attributes such as university degrees and social security numbers.

2.1.1 The Need to Protect Identity

An individual exists in many contexts, for example, email accounts, social networks, online games, government identities, work, and education. In each of these contexts, the individual has an identity; thus, an individual has many identities. There is a

growing pressure on individuals to keep these contexts from being linked together. Whenever the identities of an individual, at different contexts, are linked together, the actions that the individual takes at these contexts are also linked. The more identities are linked, the more actions are linked, which results in a profile of that individual. Profiles put the privacy of individuals at risk.

There is another reason that shows the need to keep contexts separated. The main requirement of some contexts is to be unlinkable to to other contexts. For example, Second Life is a multiplayer online game. The game allows players to interact with each other, in a virtual society. Individuals play the game to escape from the real world (one context) to be part of a virtual life (another context). Many bloggers want their identities as bloggers to be unlinkable to their identities as individuals, for various reasons, such as security. Members of certain clubs and societies may need to keep their membership at these clubs unlinkable to other contexts, say their career.

2.1.2 The Ease of Connecting Contexts

Attributes are meaningful in one or more contexts. A university ID is meaningful in a university context, while a passport number can be meaningful in a government context. Names are meaningful in both contexts. Thus, two identities, in two contexts, may get connected if they share some attributes. For example, the name of an individual on his student card is the same on his passport, which can be used to connect the two identities together.

Since many individuals have the same name, nationality, or university degree, many do not consider these attributes as part of an identity. Nevertheless, given values for only a few of these attributes may be enough to uniquely identify each

individual. Recent advances in data-mining techniques make it feasible to identify individuals based on attributes that are normally considered non-identifying. Almost every attribute is, to some degree, identifying. This constitutes a major privacy threat because individuals cannot be involved in many interactions without being profiled.

Social networks makes it easy to connect different contexts. For example, in 2009, the wife of the British MI6 chief uploaded personal information about her family, including house location and photos [9].

2.2 Identity Management

An identity-management system (IMS) facilitates the creation, storage, retrieval, and usage of individuals' identity information to authenticate and authorize those individuals at organizations. Early IMSs were designed to allow organizations to mange the identities of their users. This type of IMSs is referred to as the silo model [48], since each IMS manages the identities of a group of users in a centralized fashion. As the Internet became more popular, the number of web services and organizations an individual dealt with increased dramatically. This meant that individuals were required to keep track of a large number of partial identities, including a large number of passwords to authenticate at organizations. This phenomenon degraded the usability of the silo model, and motivated the development of a more flexible IMSs.

Instead of having one IMS per organization, a group of organizations may use one IMS, in which one organization acts as an identity provider for the users of all participant organizations. This represents the second generation of IMSs, which is still centralized, but minimizes the number of partial identities per user. An example IMS of this approach is the Microsoft Passport model [65]. A big disadvantage of

the centralized approach is that the identity provider of an IMS represents a single point of failure for that IMS. Further, many organizations may not trust the identity provider. These two disadvantages prevented Microsoft Passport from becoming a universal IMS.

Federated Identity-Management Systems (FIMSs) take a decentralized approach to identity management. Instead of having one identity provider, FIMSs enable a set of organizations to create a federation, and exchange the identity information of users. Users of one organization may authenticate at that organization; then access not only the services at that organization, but also the services at other organizations, within the federation. Some well-known FIMSs are Liberty Alliance [55], Shibboleth [74], WS-Federation [83], CardSpace [24], and sxip [79].

2.2.1 Applications of Identity-Management Systems

Identity-management systems are essential components in access control systems, as they are needed to manage user identities. Therefore, IMSs have many applications, for example, e-government, e-commerce, and grid and cloud computing.

Governments offer services to their citizens over the Internet. Citizens receive digital identities from their governments. These identities are needed to authenticate the citizens. IMSs allow governments to manage their citizen identities. Similarly, e-banks and e-commerce require IMSs.

In grid and cloud computing, the Internet is viewed as a web of resources. Organizations allow users of other organizations to access their resources. For example,

researchers from one university perform experiments using the labs of another university. Organizations may also provide the software and infrastructure to other organizations. For example, Amazon provides storage and computing services to other organizations, via Amazon's Simple Storage Service (S3) and Elastic Compute Cloud (EC2) [5]. Those organizations which provide the services require IMSs to manage the identities of their customers. The expected market size of grid and cloud computing, which is estimated by Merrill Lynch to surpass $100 billion [43], is a motivation for IMSs.

2.2.2 Privacy-Preserving Identity-Management

Privacy-preserving/enhancing identity-management aims to maximize the control that individuals have over their identity information and to minimize the identity information that individuals have to release to the system [41]. Anonymous credential systems and user-centric systems are example privacy-preserving identity-management systems. Section 2.3 review these systems.

2.2.3 Federated Identity-Management Systems (FIMSs)

FIMSs allow individuals to authenticate at identity providers once; then use multiple service providers without the need to re-authenticate at each provider. There are three main components in a FIMS: individuals, identity providers (IP) or (IdP), and service providers (SP) – also called relaying parties (RP). IPs create and certify identities for individuals, whereas SPs verify the identities and provide individuals with access to services.

Liberty Alliance is a consortium that includes Sun, HP, General Motors, and many

other global corporations. The consortium's mission is to provide open standards for the development of federated identity-management systems [61]. The architecture advocated by Liberty Alliance has five frameworks: the Identity-Federation Framework (IDFF), the Identity Web Services Framework (IDWSF), the Identity-Services Interface-Specifications (IDSIS), the Identity-Governance Framework (IGF), and the Identity-Assurance Framework (IAF). IDFF, published in 2002, provides a single means for sign-on for individuals, and simple session management. This allows individuals to sign-on at their organizations, yet be able to access services at other organizations. IDFF enables the organizations to exchange the sign-on information required to achieve the above.

IDWSF, published in 2004, builds-on IDFF to provide support for identity-based web services, such as calendars, blogs, instant messaging, and many social-networking applications. IDSIS builds on both IDFF and IDWSF to support networked identity applications, such as e-wallets.

IGF, published in 2007, permits identity and service providers to govern the process of identity information usage and dissemination. With the IAF, published in 2008, identity and service providers may assign and determine assurance levels to identity information. This helps in determining the risk of accepting identity information.

Shibboleth from Internet 2 is an open-source IMS for single sign-on across organizations. Shibboleth's approach for federation is quite similar to Liberty Alliance, but with more focus on educational institutions. Liberty Alliance and Shibboleth use the Security Assertion Markup Language (SAML) [71] for securing the communication of identity-related information among individuals, IPs and SPs. SAML 2 extends

SAML 1.1 to include the Liberty Alliance IDFF and Shibboleth. Therefore, Liberty Alliance 2 and Shibboleth 2 use SAML 2 as a basis for secure communication and for basic identity services, such as single sign-on [61].

WS-Federation is part of Web Services framework (WS) by Microsoft and IBM. WS-Federation defines how identities can be federated among different providers. While Liberty Alliance uses SAML, WS-Federation utilizes XML digital signatures standards. The WS framework has published an extension to allow WS-Federation to be interoperable with systems that uses SAML, like Liberty Alliance.

2.2.4 How Does Identity Federation Work?

Identity federation begins with the creation of a federation between a set of organizations. Whenever an organization is visited by an individual that has an account at another member of the federation, the organization gives that individual the choice to federate her identity between the two organizations. If the individual approves, each organization generates a pseudonym and associates it with the individual's account at that organization (a new account is created if it does not exist). Then, both organizations exchange the pseudonyms with each other. Whenever that individual revisits any of the two organizations, she only needs to authenticate to one of them to use the services of both.

Figure 2.1 shows a typical usage scenario in Liberty Alliance, Shibboleth, and WS-Federation. When an individual visits an SP, the SP redirects the individual to her IP's site. The IP authenticates the individual and redirects her to the SP, where the redirection message contains a proof that the individual has been authenticated. If the individual navigates from the SP to another SP, within the same federation,

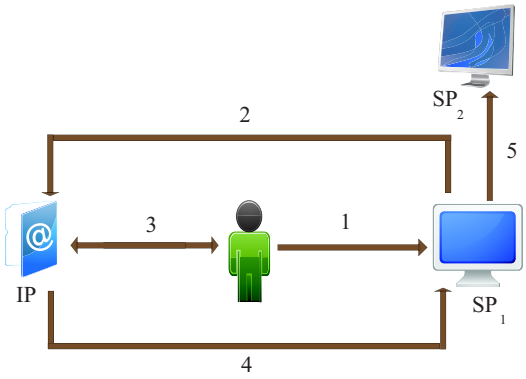

Figure 2.1: Federated identity-management systems

the individual is not redirected again to her IP for authentication. The SPs exchange the authentication information of the individual.

2.3 Related Work on Identity Management

We identify four important research topics that are closely related to this thesis. The following sections describe these topics.

2.3.1 Anonymous Credentials

In FIMSs, identity providers may supply individuals with anonymous credentials. Anonymous credentials allow individuals to interact with service providers in an anonymous fashion. However, since the pseudonyms in these credentials do not change, the actions of an individual are linkable to each other.

To overcome the linkability of the actions of an individual, Anonymous Credential Systems (ACS) allow individuals to prove the possession of credentials, without showing them to organizations. This enables an individual, for example, to visit a service provider, multiple times, while that provider is unable to link the individual's

actions. U-Prove [82] and Idemix [22] are examples of anonymous credential systems. These systems utilize zero-knowledge proofs [38] to enable individuals convince SPs that those individuals possess credentials, granted by IPs, without disclosing the actual credentials. Zero-knowledge proofs, introduced by Goldwasser *et al.* [38], are proof strategies with the zero-knowledge property. The zero-knowledge property allows a prover, who knows a correct solution for a problem, to convince a verifier that she knows that correct solution. The verifier learns nothing beyond the solution's correctness.

Originally developed by Credentica, U-Prove has been recently acquired by Microsoft. U-Prove focuses on security and privacy aspects of identity management. Idemix, developed at IBM, not only targets privacy, but also tackles the problem individuals sharing their credentials. Idemix achieves multishow unlinkability of credentials; that is, an individual may interact with same SP multiple times, without the SP being able to link these interactions together. Prime [23] is a project funded by the European Union and several corporations to support privacy-enhanced identity management. European regulations regarding privacy are the core requirements being incorporated into Prime. Prime uses Idemix protocols for issuing and verifying credentials.

A typical usage scenario in U-Prove and Idemix goes as follows (refer to Figure 2.2). Whenever an individual desires to authenticate at an SP, the individual uses a credential obtained previously from her IP. The system uses zero-knowledge proofs to convince the SP that the individual has a valid credential from the IP. Note that identity providers are not required to be online, while individuals are interacting with service providers.

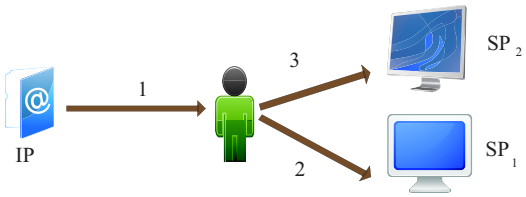

Figure 2.2: Anonymous credential systems

2.3.2 User Centric Identity Management

In user centric identity management, the focus is on individual's perspective [18]. User centric IMSs put individuals in control of what identity information is released from identity providers, and how this information is shared among service providers, within a federation. Individuals can also specify the conditions under which their identity information is allowed to be released. Spantzel *et al.* [14] list an elaborated set of requirements for user centric IMSs. CardSpace, Higgins, sxip, and OpenID [68] are examples of user-centric IMSs.

CardSpace, from Microsoft, is based on WS-Federation. CardSpace focuses on maximizing the ease of use and the individuals' control over their identities. This is done by allowing the identities, called InfoCards, to be managed at the individuals' machines. Info cards do not store identity information. Instead, these cards point to the identity providers from which the identity information can be requested. For a service provider to access identity information, the corresponding info card must be handed to the identity provider. Thus, individuals need not to worry about protecting their identities, while at the same time, they control which service providers get to use which info cards, and at which conditions.

OpenID [68] is a URL / XRI-based IMS. In such IMSs, individuals' credentials are simply URL or XRI addresses. Each address points to a document describing the

identity provider of that individual and retrieval method of the individuals' identity information. Whenever a service provider authenticates an individual, the individual supplies her id to that service provider. The service provider uses a service discovery protocol to communicate with the identity provider, and redirects the individual to the identity provider for authentication.

The Higgins Framework [80] is an open source IMS. Higgins is designed to work across many popular identity technologies, like SAML and InfoCards. It presents an application interface that allows for the integration of Higgins with other IMSs. Therefore, Higgins is compatible with many IMSs, like CardSpace, Liberty Alliance, and OpenID.

2.3.3 Interoperable Identity Management

The motivation behind designing interoperable IMSs are twofold. From an individual's perspective, interoperability enhances usability. Individuals who use one IMS can access services that implement other IMSs. From a service provider's perspective, interoperability removes the necessity of implementing a heavy-weight interface for every possible IMS. Therefore, interoperability of IMS can save governments, banks, and health institutions, time and effort of developing interfaces to their services.

There are three dimensions of interoperability: technical, informal (social / cultural), and formal (legal / organizational) [42]. In each dimension, there are many obstacles that hinder the realization of IMS interoperability. The technical dimension includes the need to resolve the difference in syntax and semantics of identity attributes among the various IMSs. The low-level representation of identities, along with the protocols required to exchange these representations, are also important

technical issues. The social and cultural dimension includes implications of interoperability on privacy. Interoperability facilitates the sharing of individuals' identity information across different organizations. Individuals consider this a potential threat to their privacy. The legal and organizational dimension deals with issues, such as the tendencies of organizations to push for their solutions, and the legal expectations of the entities, and the legal consequences of abuse across IMSs.

Higgins and OpenID tackle interoperability on the technical level by building interfaces that allow different IMSs to interoperate. The WS framework presents the WS-Interoperability framework. The framework allows SAML tokens to be translated into XML signatures and vice versa.

Paci *et al.* [66] present a protocol that uses lookup tables and ontologies to resolve the problem of naming heterogeneity of attributes. Naming heterogeneity issues handled are the use of different names to refer to the same attribute, the use of different spelling (Credit_Card *vs.* CCard), and the use of the same name, in different domains, to refer to different attributes.

There are also many projects initiated and funded by the European Union. For example, the Future of Identity in the Information Society (FIDIS) project [27] presents a framework that tackles interoperability along the three dimensions. FIDIS identifies a set of requirements for an interoperable IMS. It is based on interviews with experts from institutions involved in e-government, e-health, and e-commerce.

2.3.4 Risk and Identity Assurance

Risk management in identity management is an important topic, since handling identity information affects many stakeholders. Peterson [67] presents a set of risk metrics

that helps in determining the confidence in an IMS. Another application of risk management is in identity assurance. Identity assurance refers to the level of confidence that organizations may have in the credentials issued by identity provider [58]. A model for federated identity assurance is presented by Madsen *et al.* [58]. The model associates each level of assurance with a set of policies, which needs to be satisfied for a credential to gain that level. The policies describe the strength of the credential (password, or certificate), the identity provider's process that verifies the eligibility for that credential, and the identity provider's process that generates and manages that credential. The model requires an auditor to monitor the identity provider compliance with the policies of each assurance level. If the identity provider processes and the credential under question meet the policies of an assurance level, individuals who hold such a credential are allowed to access the services that require that level of assurance.

2.4 Limitations of Related Work

Some of the problems with FIMSs (excluding anonymous credential systems) are:

1. FIMSs allow 'non-identifying' information to be exposed to SPs. Using data-fusion techniques, this information can be merged to discover complete identities.

2. FIMSs employ privacy policies. It only takes one ill-designed policy to undermine the security and privacy of a whole system. There are many incidents that involve the disclosure of millions of credit cards, due to inappropriate decisions [3, 77].

3. FIMSs require IPs to be available whenever individuals interact with SPs. This is an onerous quality-of-service requirement. Further, IPs may build profiles of the actions of individuals, since SPs need to contact IPs and verify individuals credentials.

Anonymous Credential Systems (ACS) avoid the above problems. ACS do prove the possession of certified attributes, rather than releasing them. ACS do not rely on privacy policies as FIMS. ACS do not require IPs to be online, while individuals are interacting with SPs. However, ACS do not address the following issues.

1. *Unlinkability for group interactions.* ACSs allow an individual to perform unlinkable actions, but they fail to guarantee unlinkability in the case of group interaction. If a set of individuals, connected by a set of relations, use current ACS to prove the relations to an organization, their actions become linkable.

2. *Anonymity and unlinkability for organizations.* ACSs assume that organizations are willing to reveal their identities. In some B2C scenarios, like the "name your price" feature by Priceline, organizations sell the surplus of products and services anonymously, with the help of arbitrageurs. Since businesses cannot reveal their identities to consumers to verify the offers, arbitrageurs are needed. Businesses are being charged by arbitrageurs to act as guarantors.

3. *Unlinkability of individuals' feedback.* Individuals provide reputation systems with feedback regarding their interactions with organizations. ACSs do not incorporate the notion of reputation management. Thus, the ratings submitted by an individual can be linked to each other, and may help re-identify that individual. This discourages individuals from rating organizations.

The following chapters construct a system that provides the above functions. Chapter 3 and 4 describe the system in detail. Chapter 5, 6, 7, and 8 present the applications of the system in e-commerce, access control, reputation management, and cloud computing, respectively.

2.5 Chapter Summary

This chapter provides the reader with the required background knowledge on identity management. Section 2.1 begins by defining the term *identity* and explaining the different types of identity attributes. Section 2.2 shows the evolution of identity-management systems (IMSs) from the silo model to the federated one. The section also describes a set of popular FIMSs from the industry. Section 2.3 surveys the related work to this thesis. The work is categorized into four research topics. In each topic, the section surveys a collection of papers. Section 2.4 discusses some of the limitations of the related work.

Chapter 3

Secure Anonymous Interactions with Personas

The goal of this dissertation, as stated in Section 1.3, is to enable individuals to interact with organizations, without allowing the organizations to link their interactions together. This should be achieved not only for interactions that involve one individual, but also for interactions that involve a set of individuals connected by structured relations.

A system that achieves this goal needs constructs that allow individuals to participate in unlinkable interactions. We develop such constructs and call them personas [44, 45, 76]. Personas are *artificial identities* that assure web services that there are other organizations guaranteeing the individuals. The chapter provides a high level description of personas and the proposed system. Section 3.1 defines personas and shows their features. The architecture and components that comprise the system are presented in Section 3.2. Section 3.3 presents a more detailed description of the operations that each component performs. The threat model of the system is discussed

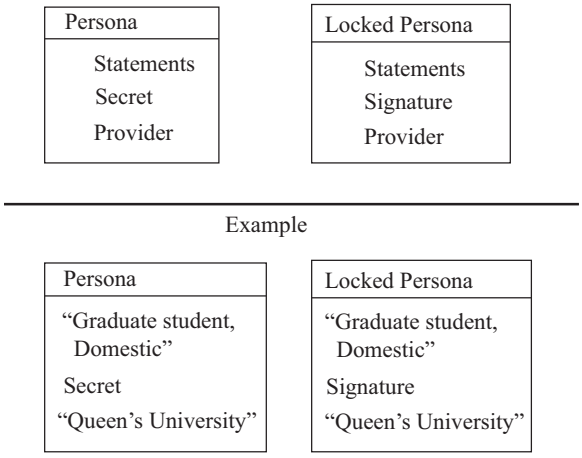

Figure 3.1: The structure of personas and locked personas

in Section 3.4.

3.1 Personas: Definitions and Features

A persona is a set of statements, where each statement asserts the status of a set attributes of an individual. Let A denote the space of attributes describing an individual. Then, $P = \{S_i(\acute{A}) : 1 \leq i \leq n, \acute{A} \subset A\}$, represents a persona with n statements, where $S_i(\acute{A})$ is the ith statement regarding a set of attributes in A.

The statements of a persona can be self-issued or certified by an organization or authority. An example of self-issued persona is an individual describing her preferences and tastes, whereas an example of an organization-issued persona is a credit card.

A persona can be used to generate a set of locked personas. Each locked persona is a proof of ownership of that persona. One may compare a locked persona to a digital signature on a message. The signature is a proof that the signer is the owner

of the private key.

Figure 3.1 shows the structure of both personas and locked personas. A persona has a statements part. Each statement specifies the status of an attribute or a set of attributes. The space of attributes is include any kind of statement that can be represented as a textual string. However, one should be careful while specifying the statements. The statements should not contain information unique to a very small set of individuals, since this may be used to re-identify those individuals. A persona also has a secret and a provider parts. The secret is used by the individual to use the persona, while the provider part specifies the organization which guarantees this persona.

A locked persona contains the statements of a persona. It has a signature on a message that serves two purposes. First, it ties the locked persona to a specific message. For example, the message can be the details of the interaction between the individual and the organization. Second, it proves that the individual has a persona.

The figure shows an example persona generated by Queen's University and sent to a student. The student may use the persona to generate locked personas and interact with organizations. Each locked persona conveys the statements of the persona, yet without showing the persona or the secret to the organization.

The following lists the properties of locked personas. The properties help achieve privacy for individuals, while guaranteeing for organizations that they are protected against misuse.

- A set of locked personas generated by a persona are unlinkable to each other. Thus, the interactions of an individual cannot be profiled. This is achieved using cryptographic constructs. Chapter 4 shows how personas are implemented and

how locked personas are generated such that each locked persona is unlinkable to other locked personas.

- A locked persona is a proof that an individual has requested a service from an organization. Usually, individuals send access requests to organizations, while these organizations reply with responses.
- A locked persona may encode an interaction details. For example, in e-commerce settings, a locked persona may encode the information on the purchased goods and time of purchase.
- A locked persona can be traced, in case of conflict resolution, to the persona which generated that locked persona. The tracing functionality is available to trusted organizations only, responsible for law enforcement.

The anonymity attacks described in Section 1.2 are based on linking similar attributes. Asserting the exact values of attributes, therefore, should be avoided, since it leads to profiling individuals.

There are four entities that comprise the system:

1. **Individuals**. Individuals use personas to protect their privacy.
2. **Persona providers (PPs)**. PPs provide individuals with personas and act as guarantors of these personas.
3. **Service providers (SPs)**. SPs offer services to individuals based on their personas.
4. **De-anonymization authorities (DAs)**. DAs trace, with the help of PPs, personas to their corresponding individuals.

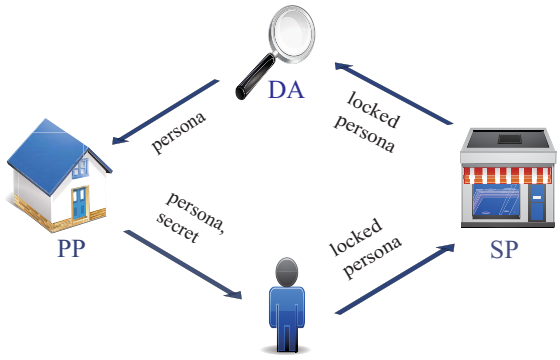

Figure 3.2: The entities generating and using personas

Figure 3.2 shows the flow of personas among the different components. A persona is generated at a PP and sent, along with a secret, to an individual. The individual generates a locked persona and sends it to an SP. The SP verifies the locked persona in a completely standalone way. If there is a problem with the transaction, for example the individual is trying to rob the merchant in some way, the SP can send the locked persona to a DA. The DA extracts the persona, and sends it to the PP. The PP is the only component that can reconstruct the mapping of the persona back to the individual. An e-commerce example is used to illustrate the entities, in which an individual shops online.

An individual requests a persona provider to generate a persona that attests the individual financial ability. Based on the individual attributes, the PP provides that individual with a persona, while keeping a record of the association of individual and persona, in case it is ever needed for de-anonymization. A secret key is also generated and sent to the individual.

Each time the individual visits a service provider, the individual uses the persona and the secret key to generate a *locked persona*. The individual sends the locked

persona to the SP. Locked personas are proofs that individuals have personas from PPs. Locked personas convey the same guarantees as personas. Each of these locked personas, however, looks different and cannot be associated either with the individual or with each other. Since locked personas are unlinkable to each other, the actions of the individual across interactions are unlinkable.

The SP verifies the information in that locked persona, without the need to contact the PP. Note that the locking process can incorporate extra information, for example, the name of the SP, the name of the product, the date and time of the interaction. Note that an individual's locked persona and an SP's reply are a proof that an individual has requested a service from the SP. This allows the interaction to be binding for the individual and the SP. PPs may also issue personas that individuals can use them no more than n-times.

Personas also offer three important features. The following sections explains these features.

3.1.1 Encoding Relations among Individuals

Personas have the ability to encode relations among individuals, which allows access policies to be formulated based on relations [45]. When the personas are presented to an SP, the SP may verify the relationships among these personas. Yet, all showings of these personas are unlinkable. If these personas are presented again to the same SP, the SP cannot tell whether these personas have been presented before.

For example, a couple may have two personas linked with the '*couple*' relationship. When the two personas are presented simultaneously to an SP, the SP can verify that the two individuals behind the personas are a couple. If the couple visit the SP at a

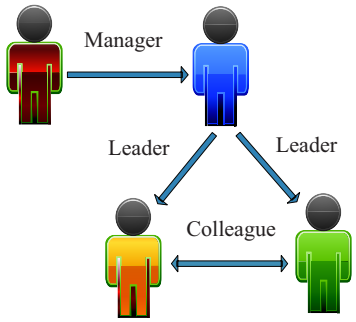

Figure 3.3: Tree-like relationship among project members

later time, the SP cannot distinguish this couple from other couples.

The same feature may encode more complex relationships, for example encoding relationships involving many individuals. The individuals involved in a more complex relationship can create related locked personas which they can present to an SP. The SP can verify the relationship among these individuals. If the individuals revisit the same SP, they cannot be distinguished from any other group of individuals having a relationship with the same structure.

This feature can be used to require that a specified set of people in a specified relationship must all participate in an action at once. For example, using a business account can require the simultaneous action of several specified company officers. Allowing a child to cross a national border can be permitted only in the presence of at least one of her parents.

Figure 3.3 illustrates a scenario where the relationship among project members has a tree-like structure. If the individuals are considered as nodes and the relations as edges, then we represent arbitrary graphs.

Although some identity-management systems (IMSs) [10, 22] may encode relations as extra attributes, this leads to a serious privacy violation. Consider the following

scenario, at which a student and her supervisor want to prove, to a university web service, the student/supervisor relation. Identity providers, in existing IMSs, may provide the student and the supervisor with two certificates. Each certificate has an attribute stating part of the relation. But that creates the problem of preventing other students from claiming that they are supervised by that supervisor. The identity provider may assign the attributes as *supervisor_pseudonym* and *student_pseudonym*, where the pseudonym parts are equal. This, however, makes the actions of the individuals linkable, since the pseudonyms remain fixed for all actions. Personas encode relations without creating this form of linkability, as discussed in Section 3.1.4.

3.1.2 Symmetric Interactions

There are many business-to-consumer (B2C) scenarios, in which service providers need to interact anonymously with customers. For example, many hotel chains use arbitrageurs to sell their room surplus at a lower price, but without necessarily revealing their brand so as not to undercut their full-price sales. To prove the quality of the services, these hotels must prove their properties, for example the star rating and the presence of amenities, such as swimming pools.

Since hotels do not reveal their brand names, they rely on arbitrageurs to prove hotel properties to customers. Well-known arbitrageurs facilitating such transactions include Priceline, Hotwire, Travelocity, and Lastminute. Arbitrageurs present individuals with offers and their properties, and reveal the hotel identities only after the transaction is complete. This leaves individuals, who wish to verify available offers, with no option other than to trust the arbitrageurs. Arbitrageurs charge the real suppliers not only for providing a service but also for acting as guarantors of product

attributes.

Personas generalize the notion of anonymity to include service providers. The goal is to make interactions between service providers and individuals symmetric, where each one proves to the other certain attributes, while both remaining anonymous.

3.1.3 Constrained Interactions

Apart from individuals' identity attributes, service providers may need to enforce a constraint on the rate at which individuals access services. For example, a reputation management system may permit an individual to submit no more than one reputation score for a given product, per time period. A provider may allow an individual to use a service n number of times per day.

Personas permit the enforcement of such constraints using *tickets*. A ticket is a message signed by a persona provider. A ticket contains the number of requests that an individual has made for a specific service, in a specific period of time. Such information helps service providers to enforce constraints that deal with the rate an individual may access a service.

Assume that a service provider allows an individual to access a service no more than once a day. To enforce this requirement, the SP requires individuals to generate locked personas and to contact their PPs to generate tickets. Individuals submit their locked personas and their tickets to the SP. The SP uses the locked personas to decide whether the individuals are eligible, and the tickets to decide whether the tickets satisfy the constraint, that is, using a service once a day. Note that although PPs generate tickets for individuals, they cannot learn the identity of the SPs that these tickets will be used at.

3.1.4 Do Personas Solve the Problems Stated in Section 1.2?

The first problem is profiling individuals by data-fusion techniques. This problem has two sides: the linkability problem, in which an individual's actions are associated with each other to create a profile, and the de-anonymization problem, in which a profile is used to re-identify an individual with the help of information from other sources. It is clear that by avoiding the linkability problem, then the de-anonymization becomes infeasible.

Personas avoid the linkability problem by conveying no more than the minimal information needed to allow individuals to participate in interactions. Since an individual uses personas to generate locked personas which are unlinkable to each other, an SP, or a set of SPs, can neither link the actions of an individual to each other nor to the individual.

The second problem occurs when an individual interacts with two organizations, possibly more, without knowing that they are partners and may share his identity information. Personas solve this problem by avoiding the linkability problem.

The third problem is the effects of human error. Assume that the system administrators at two SPs, say SP1 and SP2, set access policies such that intruders can access the SPs' database of customer information. Suppose that SP1 relies on privacy policies, while SP2 relies on personas. Intruders, accessing SP1's database, will compromise valuable identity information, like credit cards. Intruders, accessing SP2's database, will compromise locked personas, which has no value for those intruders. Each locked persona is a proof that an individual, certified by a PP, has requested a service from that SP, possibly at a specific time. There is no value for the intruder in stealing locked personas.

The fourth problem is the extension of anonymity to include the case where a set of individuals interact with an organization. The set of individuals should be capable of proving a set of relations among themselves. As shown in Subsection 3.1.1, current systems do not handle this type of interaction. Chapter 4 provides the required cryptographic constructs to achieve this property.

The fifth problem is the need of some organizations to interact, anonymously, with individuals, without reliance on arbitrageurs. Personas allow for this arrangement as described in Section 3.1.2. Chapter 5 presents a system based on personas that enables businesses to sell their products and services to consumers, anonymously, without total reliance on arbitrageurs.

The sixth problem is that current work on privacy rarely considers reputation-based trust management. Chapter 7 presents the application of personas in reputation-based trust. The chapter designs an anonymous reputation-management system.

3.1.5 Are Personas Prone to Behavioral Mining?

Behavioral mining can be used to study the patterns individuals follow in their actions to learn new information about these individuals. Since locked personas of an individual are unlinkable to each other, the actions of that individual are unlinkable to each other too. Even if an individual follows the same pattern when accessing a service, it is highly unlikely that behavioral mining can link the actions of that individual. This is true in most applications, since the set of individuals who share the same behavior is large.

3.2 System Architecture

This subsection explains each component in more detail.

3.2.1 Persona Providers, PPs

Individuals contact PPs and submit claims, while PPs verify the claims and reply with personas. Personas are generated as follows.

An individual contacts a PP and claim a set of attributes. The PP validates the individual claims. The process of validating the claims is dependent on several issues, for example, PP policies and the type of attributes claimed. Normally, this process is considered out of the scope in the majority of the work in this area. Nevertheless, when an individual claims some attributes, one of the following three scenarios is most likely to be the case.

1. The attributes are directly verifiable by the PP. In this case, the PP proceeds with validation.

2. The attributes are not directly verifiable by the PP, but the attributes are backed up by another PP. In this case, the PP requests the individual to submit a persona from the latter PP before proceeding with validation.

3. The attributes are not verifiable by the PP. In this case, the PP requires the individual to be present or to call an agent.

This is quite similar to how authentication works in the real world. All attributes are derived from a few basic identity attributes. For example, credit cards are backed up by credit ratings, which are backed up by bank balances, which require government

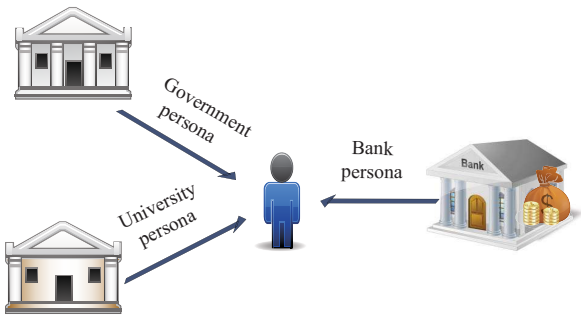

Figure 3.4: PPs providing an individual with personas

identification to open. Once the attributes are validated by the PP, a persona is generated as follows.

1. The PP encodes the individual's attributes.

2. The encoded package is signed by the PP.

3. The signed package represents the persona, which is sent back to the individual.

Identity providers in our society are of a number of different types. Governments guarantee personal identities such as citizenship; schools and universities guarantee credentials such as degrees; supervisors guarantee character or performance by writing references; and financial institutions guarantee financial worth. Any of these identity providers can act as a PP. Figure 3.4 shows an individual receiving personas from different PPs.

3.2.2 Individuals

Individuals receive personas from PPs, and store these personas for later use at SPs. Individuals must keep their personas securely stored. An individual follows these

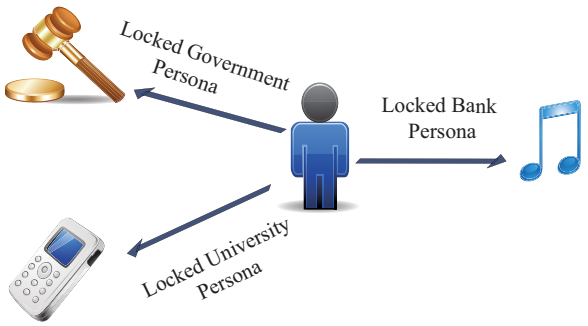

Figure 3.5: An individual using her personas at various SPs

steps to use a persona.

1. The individual requests a service from an SP, which replies with a list of required personas.

2. The individual generates a locked persona from each of the required personas.

3. The individual sends the locked personas to the SP. If the locked personas satisfy the SP's access policy for the requested service, the individual gets access.

Figure 3.5 shows the same individual using many services with her personas. The government persona is used to prove that she is over 18. She also enjoys the student offer at her phone company after providing a locked persona obtained from her university that guarantees she is eligible. She downloads music from an online store with a locked persona obtained from her bank.

3.2.3 Service Providers, SPs

Service providers can be any web service on the Internet, or indeed any real-world service provider. Each SP keeps a list of PPs that it trusts. Individuals with personas

from the trusted PPs are allowed to use the services at the SP. For example, Amazon accepts payments made using Visa and Master credit cards. The SP follows these steps to verify a locked persona of an individual.

1. The SP determines which PP parameters to use to verify the locked persona (encoded in the locked persona).
2. The SP verifies whether the locked personas are valid and the attributes satisfy the access policy of the requested service. Note that this is done without contacting the PP.
3. The SP replies with a signed message to confirm.

3.2.4 De-anonymization Authorities, DAs

DAs are invoked when there is a need to extract the persona used to generate a specific locked persona. To prevent a DA from tracing a locked persona directly to the corresponding individual, a DA can only recover the persona. Only PPs can link a persona back to an individual.

1. The DA receives the locked persona from the SP.
2. The persona behind the locked persona is extracted.
3. The DA sends the persona to the PP for further de-anonymization to an individual.

Figure 3.6 shows two scenarios involving DAs. One scenario comprises a bank, which has two components: a PP providing personas, and a DA tracing them. The

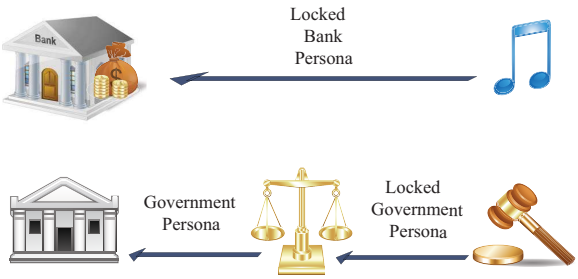

Figure 3.6: De-anonymizing personas using DAs

music store sends a locked bank persona to the bank to collect the money it is owed as the result of selling something to that persona. The bank traces the locked persona back to the individual who spent the money. The other scenario comprises an auction service sending a locked persona to a DA. The DA extracts the persona and sends it to the government to be traced to an individual.

3.2.5 Persona Use-Case

Individuals may use online resources to prepare their tax returns. For example, uFile and QuickTax are two online tools that individuals use to prepare their taxes. Once an individual prepares her return, she may print the return and mail it. She may also use NetFile to submit the return electronically. NetFile is a Canada Revenue Agency (CRA) web service that allows taxpayers to electronically submit their tax returns. NetFile is only accessible with a personalized access code. CRA provides eligible taxpayers with the needed personalized access code to use NetFile.

Currently, individuals must supply uFile or QuickTax with their information for these online tools to prepare their returns. This information includes individual names, social insurance numbers, dates of birth, addresses, postal codes, and even

their banking account information (for electronic deposit/withdrawel). Individuals pay for the services of uFile and QuickTax using credit cards. Credit card information has to be specified for uFile and QuickTax. This way of conducting business puts individual identities at risk.

Personas can be used to protect individual identities. All of the above mentioned identity information is not needed to prepare the returns, but only to associate returns with individuals. This association can be done using personas, without requiring individuals to release their identity information.

First, CRA supplies individuals with personas instead of access codes. Banks also supply individuals with personas instead of credit cards. Individuals prepare their returns at uFile or QuickTax. Instead of paying with credit cards, individuals may pay be their bank personas. Second, individuals use their bank personas to generate locked personas. uFile and QuickTax may contact banks and send the locked personas to collect their money. Individuals use their CRA personas to generate locked personas that associate returns to individuals. Third, NetFile may verify these locked personas, which shows that the individuals behind these locked personas are indeed certified by CRA to use NetFile. Finally, NetFile may de-anonymize locked personas to learn the identity of individuals behind these locked personas to finalize the tax return applications.

3.3 System Operations

This section uses a sample scenario to illustrate the operations required to manage personas. The operations are then listed and described. The descriptions are limited to show the input and output of each operation. The implementation of the operations

is left for Chapter 4.

3.3.1 A Sample Scenario

To facilitate the description of the operations, a sample scenario is used to illustrate the management of personas. In this scenario, Bob is a student at a university. He wishes to access his university library, online. Bob will be more willing to interact with the library, if the library does not build a profile of the books he is reading.

The library allows students to access the library if they show a university-issued credential. The library also has a policy that permits a student to access no more than five books in a single day.

It is clear that we have a conflict of interest. To enable Bob to have unlinkable actions at the library, his university's credential should not contain an identifying part, which tells him apart from other students. To enable the library to limit the number of books that Bob access, during a single day, the library must keep track of each book that Bob access. Current identity-management systems do not allow individuals to have unlinkable actions, while at the same time allow organizations to limit the number of times individuals access their services.

To allow Bob to have unlinkable actions at the library, while ensuring the library that students are held accountable, personas are used. The following depicts the steps that Bob, his university, and his library take to satisfy the concerns of each one of them.

The university acts as Bob's persona provider, whereas the library acts as his service provider. For simplicity, assume that the DA is the university. First, Bob requests a persona from his university. The University, after making sure that Bob is

a student, responds with **issuing a persona** to Bob. The persona acts as a university credential issued to Bob.

Bob uses the persona to access the library. He does that by **generating a locked persona** each time he wants to access the library.

The library makes sure that the locked persona corresponds to a university-issued persona by **verifying that locked persona**. Should conflicts arise, the university, which acts as the de-anonymization entity, can re-identify students by **tracing locked personas**.

The described scenario deals with Bob's concerns, however, it does not address the need to limit Bob to access no more than five books. But recall that personas allow for constrained interactions. Therefore, the library may know the number of books accessed by Bob. To do this, the library asks Bob to submit a ticket. The university **generates a ticket** for Bob, while the library **verifies that ticket**. Ticket generation does not give the university any information about the service that Bob is submitting the ticket to; that is, the university does not learn the books that Bob wants, or even if Bob is using the ticket at the library or at another place.

The bold-face words represent the operations on personas. They are described in the next section.

3.3.2 A High-Level Description

The operations are grouped by the entity which invokes them.

Operations at PPs, University

$SetupAtPP : initializations \rightarrow PP\ pparam \times PP\ prkey$

PPs, like the university for Bob, use *SetupAtPP* to generate the public parameters

PP pparam and their private keys.

$Wrap : attributes \times proof \times PP\ pparam \times PP\ prkey \rightarrow persona$

Wrap is executed by a PP to generate a persona for an individual. The PP receives a set of claimed attributes from the individual, along with the proof that individual is entitled to the attributes. The proof may take the form of a locked persona or any other forms acceptable by the PP. The PP returns a persona to that the individual. In Bob's scenario, the university executes wrap to generate a persona for Bob.

$Check\_Wrap : persona \times PP\ pparam \rightarrow boolean$

Check_Wrap is used by an individual or a PP to check if a persona is valid. Bob invokes *Check_Wrap* to check his university persona.

$Generate\_Ticket : tRequest \times locked\ persona \times PP\ pparam \times PP\ prkey \rightarrow ticket$

Generate_Ticket is executed by a PP to generate a ticket, to be used by an individual at an SP. In Bob's scenario, the university generates a ticket, based on Bob's ticket request. Bob needs a ticket whenever he wants to access a library which has a limit on the number of books accessed by Bob.

Operations by individuals, Bob

$Show : message \times persona \times PP\ pparam \times DA\ pparam \rightarrow locked\ persona$

Show is executed by an individual to generate a locked persona, proving the ownership of her persona. The individual then sends the looked persona to the SP. The show operation may also associate some meta-information with the locked persona, for example, an action, message, and time-stamp. We can treat *Show* as a signature on a message or an action. In Bob's scenario, Bob executes *Show* to generate a locked university persona.

$Selective\_Show : message \times persona \times PP\ pparam \times DA\ pparam \rightarrow locked\ persona$

An individual may use *Selective_Show* to show a subset of a persona to an SP. For example, the university may distribute Bob's attributes across several personas. Bob then can use each persona to prove a specific attribute.

Operations at SPs, University Library

$Verify : message \times locked\ persona \times PP\ pparam \times DA\ pparam \rightarrow boolean$

A verifying entity V, receiving a locked persona, uses *Verify* to check the validity of the received locked persona. If *Verify* is passed successfully, the SP knows two facts. First, the individual is indeed certified by a PP to use a persona. Second, the locked persona and the SP's signed reply are a proof that the individual has requested a service from the SP. In Bob's scenario, the university library uses *Verify* to check that the locked persona is valid with respect to the university.

$VerifyRelation : locked\ personas \times PP\ pparam \times DA\ pparam \times relations \rightarrow boolean$

A verifying entity V, receiving a set of locked personas, uses *VerifyRelation* to check the validity of not only the locked personas, but also the relations between them. If *VerifyRelation* is passed successfully, V knows three facts. First, the generating entities are indeed certified by a PP to use these personas. Second, the locked personas form a proof that these entities have participated in a transaction with V. Third, the claimed relations among these individuals are valid.

$Verify\_Ticket : ticket \times tRequest \rightarrow boolean$

Verify_Ticket is used by organizations to validate the tickets submitted by individuals, and make sure that these individuals are not exceeding the limit of access. In Bob's scenario, the library validates Bob's ticket and determines the number of books Bob accessed. The university cannot associate the tickets of Bob with each other.

Operations at DAs, University

$SetupAtDA : initializations \rightarrow DA\ pparam \times DA\ prkey$

DAs, like the university in Bob's scenario, use *Setup At DA* to generate the public parameters *DA pparam* and their private keys.

$Trace : locked\ persona \times DA\ public\ parameters \times DA\ prkey \rightarrow persona$

Trace is used by a DA to trace a locked persona back to a persona. In Bob's scenario, the university uses *Trace* to trace Bob's locked personas back to him.

3.3.3 Cryptographic Constructs

Personas can be supported by cryptographic systems that are capable of the following functionality.

1. Unlinkability of interaction transcripts. When an individual uses a certificate repeatedly at SPs, possibly at the same SP, the SPs cannot link the different usages of that certificate. This functionality is needed to prevent SPs from profiling individuals.

2. Encoding and verifying relations. Two or more individuals with arbitrary relations may use their certificates at SPs, possibly at the same SP, to prove these relations.

3. Supporting constrained interactions. Service providers should be able to specify a limit on the rate at which an individual may use a service, in a given time interval.

Idemix [22] achieves unlinkability of interaction transcripts; however, it needs to be modified to accommodate relationship verification and for constrained interactions.

To implement personas, we use the hidden ID-based signature scheme [52]. The scheme is chosen since it allow authorities to generate certificates based on identities. Certificates can be used to generate verifiable signatures on messages. The identity of the signer is not needed to verify the signature. This prevents verifiers from linking signatures produced by a signer to each other. The functionality needed to implement personas is built using the cryptographic constructs provided by the scheme.

3.4 Threat Model

The threat model is illustrated in Figure 3.7. The figure shows the attacks that can be launched against the system. The adversaries are represented as red circles and labelled A_1 to A_4. The following describes each attack:

- **Forging Personas**. A_1 issues a persona that is valid with respect to a PP.
- **Forging Locked Personas**. A_2 generates a verifiable locked persona that corresponds to a valid persona.
- **Forging Attributes**. A_2 possesses a valid persona and uses it to generate verifiable locked personas that contain attributes not certified by a PP.
- **De-anonymizing Locked Personas**. A_3 traces a locked persona to the persona used to generate that locked persona.
- **Linking Locked Personas**. A_4 links locked personas, that have been generated by a valid persona, to each other.

Chapter 4 presents a cryptographic framework that supports persona management. The chapter also shows that personas are protected against these attacks.

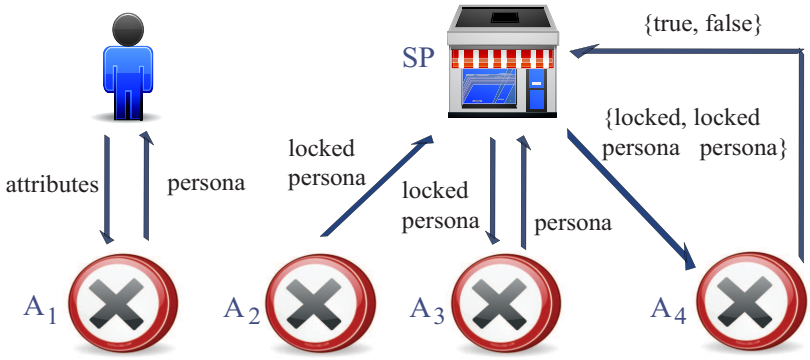

Figure 3.7: The threat model

3.5 Chapter Summary

Personas are the building block of our system. They facilitate unlinkable interactions between an individual and an organization, as well as between a set of individuals and an organization. This chapter describes personas and presents a system to manage them. The chapter serves as a high level view of the system, whereas Chapter 4 focuses on the design and implementation detail. Section 3.1 defines personas and explains their features. The architecture of the persona-management system is discussed in Section 3.2. Section 3.3 lists the operations that each entity in the system is allowed to perform. The threat model of the system is discussed in Section 3.4.

Chapter 4

A Cryptographic Framework for Personas

This chapter presents the cryptographic constructs required to implement personas and their features. The constructs uses identity-based cryptography. Section 4.1 introduces the notion of identity-based cryptography. Section 4.2 describes the Hidden Identity-based Signatures (HIDS) scheme [52]. This scheme serves as the base on which we build the required operations to support persona management. Section 4.2 also explains why HIDS needs to be modified to support personas. Section 4.3 modifies the HIDS scheme to manage personas. Appendix A describes the correctness and security of the extension, which address the threat model illustrated in the previous chapter.

4.1 Identity-based Signatures

In public-key infrastructure (PKI), an individual receives her public/private key pair from a certificate authority. In identity-based cryptography, however, the identity of an individual is the public key. The corresponding private key is generated by a private-key generator (PKG). Thus, PKGs replace certificate authorities. The advantage of ID-based cryptography is that Bob does not need to contact Alice's certificate authority to encrypt a message for her, as is the case in PKI. Bob uses Alice's identity as the public key for encryption, whereas Alice uses her private key to decrypt messages as in PKI. ID-based signatures are the signing mechanism in ID-based cryptography. Bob's PKG computes his private key and sends it to Bob. Bob uses the private key to sign messages, whereas Alice uses Bob's identity as the public key. ID-based signatures were initially proposed by Shamir [73] and early practical realization appeared in 2001 by Boneh *et al.* [17].

4.2 Hidden ID-based Signatures (HIDS)

HIDS is an identity-based signature scheme. The scheme has the following property: signed messages are verifiable without the public key (identity) of the signer. Only the public key of the identity provider is needed. The scheme employs an identification protocol and turns the protocol into a signature scheme using the Fiat-Shamir method [32]. The scheme splits the role of the identity provider into an identity manager and an opening authority. The identity manager issues certificates to individuals, while the opening authority may open the signatures generated from these certificates. Opening a signature refers to the process of extracting the public key of

the signer. The scheme provides these six operations:

`Setup`. Initializes the public/private key pair of both Identity Manager (IDP) and de-anonymizing Authority (DA).

`Registration`. The IDP registers an individual by issuing a certificate, which is a signature on that individual identity produced by the IDP private key.

`Check Reg`. The individual checks whether the identity and certificate pair are valid with respect to each other.

`Sign`. Signatures are generated as follows. The individual commits the identity and the certificate. Then, a proof of knowledge (Σ-protocol) is used to prove the knowledge of the value of the committed identity and certificate, and that the certificate is a signature on that identity. The output of Σ-protocol is hashed along with the message to be signed. The committed identity and certificate, the protocol output, the produced hash, and the message comprise the signature.

`Verify`. The verifier uses the IDP public key to check the signature is valid and that the signer certificate and identity are encrypted with the DA public key.

`Open`. The DA uses its private key to decrypt and extract the signer's committed identity from a valid signature.

The scheme provides the basic cryptographic support for generating, verifying, and tracing signatures. We extend the scheme to provide the needed persona features, such as showing attributes, proving relations, and enabling constrained interactions.

4.2.1 HIDS Operations

`Setup`: $\{public\ parameters, IDP\ keys, DA\ keys\} \leftarrow Setup.$

Setup is used to generate the required public/private keys for the IDP and DA. The keys are generated based on Boneh and Boyen [15] signature scheme. Setup generates $(p, g, \mathbb{G}, \mathbb{G}_2, e)$, where $\mathbb{G}$ and $\mathbb{G}_2$ are cyclic groups of prime order p, g is a generator for $\mathbb{G}$, and e is a bilinear map, $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_2$. The IDP public key is the pair ($X = g^x$, $Y = g^y$), where x and y are random elements in $\mathbb{Z}_p$. The IDP private key is the pair (x, y). DA public key is given by (u, v, w), where w is a random element in $\mathbb{G}$ and $w = u^b = v^d$, b and d are random elements in $\mathbb{Z}_p$. DA private key is (b, d). The public parameters of the system are $(p, g, \mathbb{G}, \mathbb{G}_2, e, X, Y, u, v, w, h, H)$, where h is a random element in $\mathbb{G}$, and H is a hash function.

`Register`: *certificate* $\leftarrow$ *Register*(*identifier*)

When an individual requests a certificate from the IDP, the IDP encodes that individual's identity as an identifier, say I. The mapping between the identifiers and the real identity of the individual is securely stored at the IDP. The IDP issues a certificate C to that individual by signing I with its private key. Note that r is a random element in $\mathbb{Z}_p$.

$$C = \{s = g^{(x+I+y\,r)^{-1}}, r\} \tag{4.1}$$

`Check Reg`: *boolean* $\leftarrow$ *Check Reg*(*identifier*, *certificate*)

An individual may check the validity of her certificate by checking if the following condition holds: $e(s, Xg^I Y^r) = e(g, g)$.

`Sign`: *signature* $\leftarrow$ *Sign*(*identifier*, *certificate*, *message*)

This operation is used to generate signatures on messages. Signing a message M with a certificate C requires these steps:

- The individual uses the encryption scheme of Boneh *et al.* [16] to commit I in

(U, V, W).

$$U = u^k, V = v^l, W = w^{l+k}\, g^I \tag{4.2}$$

where l and k are random numbers in $\mathbb{Z}_p$.

- The individual commits C in (S, R), where $S = g^{r_1} s$, $R = g^{r_2} h^{r_1} Y^r$, and r_1 and r_2 are random elements in $\mathbb{Z}_p$.
- The individual uses the Σ-protocol described below, to prove the knowledge of the committed values of C and I, and that C is a valid signature on I.
- The variables of the Σ-protocol are hashed along with the message to be signed.
- The committed values, hash, and M represents a HIDS signature.

The Σ-protocol

The individual uses the protocol to prove the knowledge of C and I committed in S and R and that C is a valid signature on I. Compute $\alpha_1 = r_1 k$, $\alpha_2 = r_1 l$, $\alpha_3 = r_1 r_2$, $\alpha_4 = r_1^2$, $\alpha_5 = r_1 r$. Choose η_I, η_k, η_l, η_r, η_{r_1}, η_{r_2}, η_{α_1}, η_{α_2}, η_{α_3}, η_{α_4}, and η_{α_5} randomly from $\mathbb{Z}_p$.

$$B_1 = u^{-\eta_k}, B_2 = v^{-\eta_l}, B_3 = w^{-(\eta_k+\eta_l)} g^{-\eta_I}, B_4 = g^{-\eta_{r_2}} h^{-\eta_{r_1}} Y^{-\eta_r},$$

$$B_5 = U^{-\eta_{r_1}} u^{\eta_{\alpha_1}}, B_6 = V^{-\eta_{r_1}} v^{\eta_{\alpha_2}}, B_7 = R^{-\eta_{r_1}} g^{\eta_{\alpha_3}} h^{\eta_{\alpha_4}} Y^{\eta_{\alpha_5}},$$

$$B_8 = e(g, X\ W\ R)^{\eta_{r_1}} e(S, w)^{\eta_k+\eta_l} e(g, w)^{-(\eta_{\alpha_1}+\eta_{\alpha_2})} e(S, g)^{\eta_{r_2}} e(g, g)^{-\eta_{\alpha_3}} e(S, h)^{\eta_{r_1}} e(g, h)^{-\eta_{\alpha_4}}$$

$$c = H(M, S, R, U, V, W, B_1, ..., B_8), \lambda_I = \eta_I + c\, I, \lambda_r = \eta_r + c\, r,$$

$$\lambda_{r_1} = \eta_{r_1} + c\, r_1, \lambda_{r_2} = \eta_{r_2} + c\, r_2, \lambda_k = \eta_k + c\, k, \lambda_l = \eta_l + c\, l, \lambda_{\alpha_1} = \eta_{\alpha_1} + c\, \alpha_1,$$

$$\lambda_{\alpha_2} = \eta_{\alpha_2} + c\, \alpha_2, \lambda_{\alpha_3} = \eta_{\alpha_3} + c\, \alpha_3, \lambda_{\alpha_4} = \eta_{\alpha_4} + c\, \alpha_4, \lambda_{\alpha_5} = \eta_{\alpha_5} + c\, \alpha_5 \tag{4.3}$$

The tuple $\sigma = \{S, R, U, V, W, c, \lambda_r, \lambda_{r_1}, \lambda_{r_2}, \lambda_k, \lambda_l, \lambda_I, \lambda_{\alpha_1}, \lambda_{\alpha_2}, \lambda_{\alpha_3}, \lambda_{\alpha_4}, \lambda_{\alpha_5}\}$ is the HIDS signature on M.

`Verify`: $boolean \leftarrow Verify(signature, message)$

The operation checks whether a received (signature, message) pair represents a valid HIDS signature. The verifier uses the following condition to check that σ is a valid signature on M.

$$\begin{aligned} c = H(M, S, R, U, V, W, U^c u^{-\lambda_k}, V^c v^{-\lambda_l}, W^c w^{-(\lambda_k+\lambda_l)} g^{\lambda_I}, \\ R^c g^{-\lambda_{r_2}} h^{-\lambda_{r_1}} Y^{-\lambda_r}, U^{-\lambda_{r_1}} u^{\lambda_{\alpha_1}}, V^{-\lambda_{r_1}} v^{\lambda_{\alpha_2}}, R^{-\lambda_{r_1}} g^{\lambda_{\alpha_3}} h^{\lambda_{\alpha_4}} Y^{\lambda_{\alpha_5}}, \\ e(g, X\ W\ R)^{\lambda_{r_1}} e(S, w)^{(\lambda_k+\lambda_l)} e(g, w)^{-(\lambda_{\alpha_1}+\lambda_{\alpha_2})} e(S, g)^{\lambda_{r_2}} e(g, g)^{-\lambda_{\alpha_3}} \\ e(S, h)^{\lambda_{r_1}} e(g, h)^{-\lambda_{\alpha_4}} (e(g, g)/e(S, X\ W\ R))^c) \end{aligned} \quad (4.4)$$

`Open`: $identifier \leftarrow Open(signature)$

The private key of the DA (b, d) is used to extract g^I from the commitment (U, V, W); then g^I is sent to the IDP ($g^I = U^{-b} V^{-d} W$). The IDP maps the identifier back to the real identity by looking up the values of g^I from a table.

4.3 The Extended HIDS Scheme

The HIDS scheme cannot be used as is to implement the needed system operations. The signer can prove the possession of a certificate from the IDP, but nothing beyond that. The scheme, therefore, needs modification to allow individuals to show attributes, prove relations, or rate service providers. We extend the scheme by modifying each operation. The modified system is called the extended HIDS for short.

In the extended HIDS, the PP plays the role of the IDP. The next section describes the operations of the extended HIDS.

4.3.1 Extended HIDS Operations

This section presents our HIDS extension to support the required functionality. The correctness and security of the extension are addressed in Appendix A. A subscript e is appended to the names of the extended HIDS operations to distinguish them from the HIDS ones.

`Setup`$_e$: $\{public\ parameters, PP\ keys, DA\ keys\} \leftarrow Setup_e$

Setup is not changed.

`Register`$_e$: $\{persona, secret\} \leftarrow Register_e(identifier)$

A PP uses HIDS's Register operation to issue certificates to individuals. When an individual requests a certificate from a PP, the PP generates two identifiers (I_{base}, and I_{full}). The difference to the HIDS identifiers is that these identifiers are composed of two parts: a *pseudonym* part, which is a random number to distinguish between individuals, and an *attributes* part, which encodes the attributes of the individual. The pseudonym part of I_{base} and I_{full} are equal. The attributes part of I_{base} is assigned to 0, that is, no attributes, while the attributes part of I_{full} is assigned to the encoding of the attributes that the individual is entitled to. For simplicity, assume that i bits of the identifier encodes the pseudonym, while the remaining j bits encodes the attributes.

The PP invokes the HIDS's Register operation twice, once per identifier. The PP sends the two identifiers and the certificate on them to the individual. The identifiers (I_{base}, I_{full}) constitute the persona, while the certificates (C_{base}, C_{full}) are the secret.

$$C_{base} \leftarrow Register(I_{base}),\ C_{full} \leftarrow Register(I_{full}) \tag{4.5}$$

$$persona = \{I_{base}, I_{full}\},\ secret = \{C_{base}, C_{full}\} \tag{4.6}$$

`Check Reg`$_e$: $boolean \leftarrow Check\ Reg(persona, secret)$

An individual may check the validity of her (persona, secret) pair by checking if the following condition holds:

$$true = Check\ Reg(I_{base}, C_{base})\ \&\&\ true = Check\ Reg(I_{full}, C_{full}) \tag{4.7}$$

$\texttt{Sign}_e$: $locked\ persona \leftarrow Sign_e(persona, secret, message, attributes)$

This operation is used to generate locked personas. This is achieved by generating HIDS signatures on messages. Signatures are the implementation of locked personas, where each (signature, message) pair represents a locked persona. $\texttt{Sign}_e$ can be used to sign messages, with or without showing attributes. For example, an individual affiliated with a university authenticates to the ACM Digital Library, which requires nothing more than that the individual is affiliated with that university. In this case, the individual supplies I_{base} to to *Sign* operation of the HIDS scheme to sign a message M, which produces σ_{base} (see Equation 4.8). The pair (M, σ_{base}) is a locked persona. The individual sends the signature and the message to the verifying entity.

$$\sigma_{base} \leftarrow Sign(M, I_{base}, C_{base}) \tag{4.8}$$

If the individual needs to show attributes to the verifying entity, both identifiers are needed. $\texttt{Sign}_e$ invokes two instances of the HIDS's Sign operation, once per each (identifier, certificate) pair. Locked personas generated by this type of $\texttt{Sign}_e$ contain attributes as well (see Equation 4.9). A locked persona is the tuple (signature, message, attribute).

$$\sigma_{base} \leftarrow Sign(M, I_{base}, C_{base}), \sigma_{full} \leftarrow Sign(M, I_{full}, C_{full})$$
$$\sigma_{base} = \{S_{base}, R_{base}, U_{base}, V_{base}, W_{base}, ...\}, \sigma_{full} = \{S_{full}, R_{full}, U_{full}, V_{full}, W_{full}, ...\} \tag{4.9}$$

Note that the same message M is used in both instances. Further, both instances of Sign should use the same values for the random variables l and k in Equation 4.2. The individual sends the signatures, the message, and the attributes to the verifying entity.

`Verify`$_e$: $boolean \leftarrow Verify_e(locked\ persona, message, attribute)$

The `Verify`$_e$ operation has two flavours: one to deal with signatures generated by Equation 4.8, and another to deal with signatures of Equation 4.9. If Equation 4.8 is used to generate a signature, then the HIDS's Verify operation is supplied with the (message, signature) pair. If Verify returns true, then σ_{base} is a valid signature on M.

$$true = Verify(M, \sigma_{base}) \tag{4.10}$$

If Equation 4.9 is used to generate signatures, the verification proceeds as follows (this is the case where an individual needs to prove attributes to the verifying entity). First, the HIDS's Verify operation is invoked twice to check the validity of both signatures (Equation 4.11). If Equation 4.11 holds, then both signatures are valid.

$$true = Verify(M, \sigma_{base})\ \&\&\ true = Verify(M, \sigma_{full}) \tag{4.11}$$

Second, the attributes that the individual is claiming are checked. Recall that the difference between I_{base} and I_{full} is that I_{full} encodes the individual's attributes, while I_{base} does not, which implies $attributes = I_{full} - I_{base}$. Recall also that σ_{base} contains the commitment of I_{base}, while σ_{full} contains the commitment of I_{full} (Equation 4.12).

$$W_{base} = w^{l+k}\, g^{I_{base}},\ \ W_{full} = w^{l+k}\, g^{I_{full}} \tag{4.12}$$

To check whether the attributes that the individual is claiming are the same attributes encoded in the identifiers supplied to her by the PP, the verifying entity

checks whether Equation 4.13 holds:

$$W_{base}\, g^{attributes} = W_{full} \tag{4.13}$$

We need to prevent an individual from swapping W_{base} and W_{full}, which would enable her to claim $(-attributes)$, instead of $attributes$. This is done by appending two bits to both identifiers at the most significant part. I_{base} bits become the binary string of 00 appended to I_{base} bits; that is, $00 + I_{base}$. I_{full} bits become the binary string of 01 appended to I_{full} bits; that is, $01 + I_{full}$. Equation 4.13 becomes:

$$W_{base}\, g^{second\_bit\_set}\, g^{attributes} = W_{full} \tag{4.14}$$

second_bit_set is a binary string with the same number of bits as I_{base}. All bits of *second_bit_set* are assigned to 0, except for the 2nd most significant bit, which is assigned to 1.

`Open`$_e$: $persona \leftarrow Open_e(locked\ persona)$

The HIDS Open operation is used to extract $g^{I_{base}}$ and $g^{I_{full}}$ from the commitments (Equation 4.15). The identifiers are then sent to the PP. The PP maps the identifiers back to the real identity by looking up the values of the identifiers from a table.

$$I_{base} = Open(\sigma_{base}),\ \ I_{full} = Open(\sigma_{full}) \tag{4.15}$$

4.3.2 Linkable Signatures

The sign operation of the extended HIDS generates two HIDS signatures, σ_{base} and σ_{full}, that are linkable to each other. The signatures serve two purposes: proving that the individual has a persona from a PP, and proving that the individual is entitled to the attributes inferred from the two signatures. Each time `Sign`$_e$ is invoked, it generates a new pair of two signatures that are linkable to each other, but are

unlinkable to other pairs. In some cases, however, an individual may need to produce a pair of signatures that is linkable to a previous one. To generate a new pair that is linkable to a previous pair, the individual must use the same message and the same values for the random variables l and k, when generating the new one.

4.3.3 Selective Release of Attributes

Instead of having two identifiers: I_{base} and I_{full}, the PP can provide an individual with many identifiers. Selective release of attributes is achieved by providing an individual with an identifier per attribute or a set of attributes. Let I_{age} be of the same bit-length as I_{full}, and the *pseudonym* bits be equal. However, all the *attributes* bits are 0s except for the bits encoding the age. An individual with I_{age} certified by PP can use the Sign operation of the extended HIDS to show the age only as follows.

$$\sigma_{base} \leftarrow Sign(M, I_{base}, C_{base}),\ \sigma_{age} \leftarrow Sign(M, I_{age}, C_{age})$$
$$\sigma_{base} = \{S_{base}, R_{base}, U_{base}, V_{base}, W_{base}, ...\},\ \sigma_{age} = \{S_{age}, R_{age}, U_{age}, V_{age}, W_{age}, ...\} \tag{4.16}$$

Recall that $W_{age} = w^{l+k}\, g^{I_{age}}$. If Equation 4.17 holds, then the individual is certified by the PP to have that *age* attribute.

$$true = Verify(M, \sigma_{base})\ \&\&\ true = Verify(M, \sigma_{age})\ \&\&\ W_{base}\, g^{I_{age}} = W_{age} \tag{4.17}$$

4.3.4 Encoding and Verifying Relations

Similar to the way personas prove attributes, personas may prove the existence of relations among individuals. An identifier is composed of a pseudonym part and an attribute part. To encode relations, a third part is added, called a *relation*. An

identifier becomes the composition of a pseudonym part, a relation part, and an attribute part.

$$identifier = pseudonym, relation, attributes$$

Let I_1 and I_2 be two base identifiers issued by a PP for individuals D_1 and D_2, respectively. Let the *pseudonym* bits of both identifiers be equal, and there be a relation between D_1 and D_2, for example, D_1 is the boss of D_2. The PP encodes that relation by giving D_1 and D_2 different values for the *relation* bits as follows. The relation bits of D_1 and D_2 are set to re_1 and re_2, respectively, such that $relation = re_2 - re_1$. Note that the *attribute* bits are assigned to 0 in both identifiers. The identifiers for D_1 and D_2 become

$$\begin{aligned} I_1 &= pseudonym,\ re_1,\ 0 \\ I_2 &= pseudonym,\ re_2,\ 0 \end{aligned} \tag{4.18}$$

The PP then register the identifiers, as in Equation 4.6, to generate a persona and a secret pair for each individual, (P_1, S_1) and (P_2, S_2).

$$\{P_1, S_1\} \leftarrow Register_e(I_1),\ \ \{P_2, S_2\} \leftarrow Register_e(I_2), \tag{4.19}$$

When D_1 and D_2 want to prove their relation to a verifying entity V, both use $\texttt{Sign}_e$ to sign a message and produce two locked personas (LP_1 and LP_2). Then, they send the locked personas and *relation* to V. Note that *relation* is appended to bits of 0s of length a (the attribute bit size).

$$LP_1 \leftarrow Sign_e(M, P_1, S_1),\ LP_2 \leftarrow Sign_e(M, P_2, S_2)$$

$$LP_1 = \{\sigma_{D_1} = \{S_1, R_1, U_1, V_1, W_1, ...\}, M\},\ LP_2 = \{\sigma_{D_2} = \{S_2, R_2, U_2, V_2, W_2, ...\}, M\} \tag{4.20}$$

The verifying entity uses `VerifyRelation` to verify the relation. The two locked

personas must be linkable for the verification to be possible; see Section 4.3.2 for information on linkable signatures.

`VerifyRelation`$_e$: *boolean* $\leftarrow$ *VerifyRelation*$_e$(*locked persona*, *locked persona*, *relation*)

This operation takes two locked personas, which represent signatures on a message, and a relation. The operation verifies each locked persona alone as in the `Verify`$_e$ operation; then Equation 4.21 is used to verify the relation. W_1 and W_2 are computed as W is computed for the case of one individual, see Equation 4.2.

$$W_1 \, g^{relation} = W_2 \tag{4.21}$$

In the same manner, we can specify relations that involve several individuals. In other words, personas can model graphs, where the nodes are individuals and the edges are their relations. We achieve this by computing an adjacency matrix for the required graph. Each cell encodes the relation between two individuals: the individual corresponding to the column of the cell, and the individual corresponding to the row. Thus, each row encodes the set of relations between an individual and the remaining individuals.

Now, we explain in details how a PP provides a set of individuals D with a set of personas P and the corresponding secrets S, allowing them to prove a set of relations R. Let D_i denotes the i^{th} individual, R_i denotes the set of relations of the i^{th} individual, R_i^j denotes the cell at row i and column j, and S_i denotes the i^{th} persona.

Algorithm 4.1 takes R as input and produces P and S as output. From R we compute $\acute{R}$, which combines the set R_i into a single value. The *relation* part of the identifier of P_i is set to $\acute{R}_i$. Then, all P_i and S_i are generated to get P and S.

Note that the *attribute* part is 0 for all P_i, and that all P_i have the same value for *pseudonym*. The algorithm above encodes the relations of D_i, $1 \leq i \leq n-1$. The

Input: R
Output: S
$\acute{R}_1 = 0$
foreach $\acute{R}_i$ *in* $\acute{R}$, $i \neq 1$ **do**
  $temp = \sqrt{(R^1_{i-1})^2 + (R^2_{i-1})^2 + ... + (R^n_{i-1})^2}$
  $\acute{R}_i = temp + \acute{R}_{i-1}$
end
$pseudonym = random$
$attribute = 0$
foreach P_i *in* P **do**
  $relation = \acute{R}_i$
  $identifier = pseudonym$, $relation$, $attribute$
  $secret = Register_e(identifier)$
  $P_i = identifier$, $S_i = secret$
end

Figure 4.1: Generating personas based on an adjacency matrix

relations of D_n can be inferred from other relations (undirected graphs). For directed graphs, a new persona P_{n+1} is needed to compensate. P_{i+1} is computed as other P_i in the algorithm. The PP finally sends P and S to D, where each D_i receives a pair (P_i, S_i). In case of directed graphs, D_n receives two pairs (P_n, S_n) and (P_{n+1}, S_{n+1}).

Now we turn to how the individuals prove R to an entity V, using P and S. Each D_i signs the same message M using P_i and S_i and sends the resulted locked persona LP_i to V, $LP_i \leftarrow Sign_e(M, P_i, S_i)$. Let LP denotes the set of the locked personas. The individuals also send R to V. The entity V runs Algorithm 4.2. It is clear from Algorithm 4.1 that the relations of the i^{th} individual can be recovered from the $\acute{R}_i$ and $\acute{R}_{i+1}$. The algorithm uses LP_{i+1}, LP_i, and A_i as input to $VerifyRelation_e$. If any instance of $VerifyRelation_e$ does not pass, the algorithm outputs reject. Otherwise, it outputs accept.

Input: R, LP
Output: *accept*, *reject*
foreach LP_i *in* LP **do**
  $relation = \sqrt{(R_i^1)^2 + (R_i^2)^2 + ... + (R_i^n)^2}$
  if $VerifyRelation_e(LP_{i+1}, LP_i, relation) = reject$ **then**
    output reject
  end
end
output accept

Figure 4.2: Verifying locked personas against an adjacency matrix

4.3.5 Ticket Management

The following describes the generation and verification of tickets.

`GenerateTicket`: $ticket \leftarrow GenerateTicket(ticket\ request)$

An individual contacts her PP to generate a ticket to be used at an SP. The individual prepares a ticket request m_I and sends it to her PP.

$$m_I = \{h_1, h_2\},\ h_1 = \tilde{\mathcal{H}}(lPersona),\ h_2 = \tilde{\mathcal{H}}(i, PP, SP) \tag{4.22}$$

where $\tilde{\mathcal{H}}$ is collision-resistant hash function, i is the time interval from the SP perspective, and *lPersona* is the locked persona used to interact with SP. Since h_1 and h_2 are hash values, the PP cannot determine SP's identity. The PP does not know for which SP the ticket is generated.

The PP records the total number of times h_2 has been submitted by the same individual, in the current interval i_{PP}, which may or may not be the same as the interval i. The total n is incremented and appended to m_I to get m_{PP}. The PP generates a ticket t by signing m_{PP} with PP's key *PP key*. (any public cryptography

algorithm, for example RSA, could be used).

$$t = \{m_{PP}, s\},\ m_{PP} = \{h_1, h_2, n, i_{PP}\},\ s = \mathcal{S}_{PP\ key}(m_{PP}) \tag{4.23}$$

where $\mathcal{S}$ generates signatures based on the key *PP key*. The PP sends t to the individual as a ticket.

`VerifyTicket`. The individual forwards h_1, h_2, i, $lPersona$, SP, PP, and t to the SP. The SP verifies and evaluates t against the constraint attached to the service requested by the individual. If the following equality holds, the SP is ensured that t is indeed generated by PP.

$$h_1 = \tilde{\mathcal{H}}(lPersona) \qquad \&\& \qquad h_2 = \tilde{\mathcal{H}}(i,\ PP,\ SP)$$
$$true = \mathcal{V}_{PP\ key}(s, m_{PP}) \tag{4.24}$$

where $\mathcal{V}$ verifies signatures based on the key *PP key*.

Finally, the SP checks whether $n \leq threshold$, where *threshold* is the limit set by the SP per individual, in the interval i.

The described algorithm enables SPs to put additional constraints on the rate or way individuals access services, while it prevents PP from knowing which SPs are being used. Section 7.4.2 applies the notion of constrained interactions in the area of anonymous reputation management.

PPs may compute h_2 for commonly used services and products; and use such values to query individuals' requests to determine which individuals have used these products. We assume that PPs do not carry out such attacks. Note that one may use a non-collision resistant hash function, but this leads to collisions. In this case, SPs may deny individuals receiving legitimate access to services.

Table 4.1: Mapping cryptographic constructs to the system operations

Concepts and Operations		The Extended HIDS
PP	→	IDP
DA	→	DA
persona	→	identifier
secret	→	certificate
attribute	→	attributes part of an identifier
locked persona	→	(HIDS signature, message)
ticket	→	tickets as in Section 4.3.5
Wrap	→	$\texttt{Register}_e$
Check_Wrap	→	$\texttt{Check Reg}_e$
Show	→	$\texttt{Sign}_e$
Verify	→	$\texttt{Verify}_e$
Trace	→	$\texttt{Open}_e$
SelectiveShow	→	Section 4.3.3
VerifyRelation	→	Section 4.3.4
GenerateTicket	→	Section 4.3.5
VerifyTicket	→	Section 4.3.5

4.3.6 System Operations: Mapping the Constructs

The building blocks of the system are now ready and given by Table 4.1. The table shows each operation/concept in our system and its equivalent construction that uses and extends the hidden ID-based signatures. `Show` and `Verify` allows for anonymity and unlinkability of interactions. `SelectiveShow` achieves selective release of attributes, whereas encoding and verifying relations is achieved by `VerifyRelation`. `Trace` implements persona traceability. `SubmitReputation` and `UpdateReputation` allow for anonymous reputation management. `VerifyTicket` and `GenerateTicket` permit ticket management.

4.3.7 Implementation

We have implemented and tested the system with the help of the Pairing-based Cryptography (PBC) library [56]. The PBC is a free library written in C and it provides the necessary functions to write programs that handle elliptic curve generation, elliptic curve arithmetic, and pairing computation. The fastest pairing operation takes 11ms on a 1 GHz Pentium 3 machine. The HIDS Sign operation can be optimized to generate a HIDS signature with two pairing operations and 14 exponentiations. The HIDS Verify operation can be optimized to verify a HIDS signature with two pairing operations and 10 exponentiations [52]. Therefore, generating a locked persona that does not show any attribute requires the same number of parings and exponentiations to generate one HIDS signature. Verifying that locked persona requires the same number of parings and exponentiations to verify one HIDS signature. To generate a locked persona that shows attributes, one need double the operations for generating a HIDS signature, that is, four pairings and 28 exponentiations. Verifying that locked persona requires four pairings and 20 exponentiations.

We used a 2 GHz Pentium III machine to test the response time and compare it to the RSA algorithm. Generating a locked persona takes 200 milliseconds, while verifying that locked persona takes 240 ms. The total is 440 ms. The RSA algorithm requires approximately 80 ms to generate and verify a signature. The reported times are based on the same machine used to test our system.

4.3.8 Limitations

The limitations of personas are:

- The system does not provide a mechanism to revoke personas. The system may

compensate for this by changing its parameters, every specific interval of time, and re-issue personas for unrevoked individuals.

- Let be P an access policy that is composed of several clauses separated by the 'OR' operator; that is, $P = p_1|...|p_n$. Let A a set of identity attributes of an individual. Let A satisfy one condition of P. Some systems allow for the evaluation of P, such that the evaluator does not learn which condition of P has been satisfied. This minimizes the knowledge that the evaluator gain from evaluating P. Our system does not implement this feature.

- Sharing personas allows an individual, who is not entitled to access a service, to have illicit access. Personas do not implement techniques to deter individuals from sharing their personas.

- Since personas are based on an identity-based signature scheme, which is based on pairing-based cryptography, personas takes more time compared to RSA. However, research in number theory is enhancing the performance of pairing operations [8].

4.4 Chapter Summary

In this chapter, the design and implementation of the persona management system are presented. The chapter presents a framework that provide the necessary cryptographic support to facilitate the management of personas. The framework is based on extending the Hidden Identity-based Signature scheme [52]. HIDS is described in Section 4.2. Section 4.2 also shows that HIDS cannot be used as is to support personas. Section 4.3 presents our extension of HIDS.

Chapter 5

Persona Applications: E-Commerce

While the majority of research on anonymity is focused on individuals, there are an increasing number of scenarios that demand anonymity for service providers as well. For example, in many business to consumer (B2C) scenarios, service providers sell their surplus to individuals for lower prices through arbitrageurs. Those service providers must remain anonymous, to avoid discouraging customers from buying directly from the service provider, at the regular price.

This chapter presents the application of personas in e-commerce. Section 5.1 states the motivation and our objectives. The related work on anonymity for service providers are described in Section 5.2. The section shows the differences of the related work to the work described in this chapter. Section 5.3 presents our approach for allowing service providers to interact, anonymously, with consumers.

5.1 Objectives

There are two issues that have remained largely unaddressed in previous work. The first is that an asymmetry is assumed between the roles of customer and supplier. Customers are required to provide a *credential* that typically corresponds to "is able to pay"; whereas the supplier participates in the transaction on the basis of "trust" or "reputation"[1]. A customer's credential is always backed by some other organization: cash (backed by a government); a credit card (backed by a credit card company or bank); or even some encrypted attribute, such as a gift card (backed by an authentication mechanism). The use of trust or reputation means that the supplier must always be identifiable. There are, however, interesting and important possibilities created by allowing suppliers to participate on the basis of credentials rather than trust or reputation.

For example, a hotel chain may wish to sell its room surplus at a lower price, but without necessarily revealing which chain's rooms they are so as not to undercut their full-price sales. To do this successfully, they must be able to prove properties of the rooms and hotels to potential customers, for example the star rating and the presence of amenities such as swimming pools. In other words, rather than rely on their *brand*, they must be explicit about the quality of the product on its own terms.

There are many business-to-consumer (B2C) scenarios of this kind. At present, the reputation issue is handled by using arbitrageurs whose role is to act as surrogate suppliers, hiding the actual suppliers but guaranteeing properties of the products. Well-known arbitrageurs facilitating such transactions include Priceline, Hotwire,

[1]Of course, there *is* an asymmetry because the customer can choose the supplier, while the supplier cannot directly choose the customer. However, while this asymmetry is strong in retail interactions, it becomes steadily weaker in business-to-business (B2B) interactions.

Travelocity, and Lastminute. When individuals browse Priceline for hotel offers in New York, Priceline supplies individuals with matching offers, but without disclosing hotel names. Since the hotel identities are not disclosed before transactions are complete, individuals have no way, other than to trust Priceline, to verify the offers. Arbitrageurs charge the real suppliers not only for providing a service, but also for acting as guarantors of product attributes.

The second issue, which arises in individual interactions but is more significant in B2B interactions, is that many kinds of transactions require a number of entities acting in a particular arrangement or structure to play the role of a single participant in the transaction. For example, some countries provide medical care to the dependants of individuals working legally in the country. Access to medical care requires demonstrating that one individual is, say, under eighteen; the other is working legally in the country; *and* there is a family relationship between them.

In another important example, an organization announces a call for tenders that requires a contractor and two subcontractors, each with specific properties. At present, a contractor can bid anonymously, or can demonstrate that the subcontractors have the required properties, but not both at once. Current anonymous-auction systems, as the one presented by Trevathan *et al.* [81], allow for anonymous bidding, but nothing beyond that. Also a subcontractor might wish to be involved in more than one of the bidding consortia to increase its chances of success, but this is impossible at present because of the social consequences.

5.1.1 Middle vs. Backend Guarantor

The objective of this chapter is to allow service providers to be anonymous, while reducing the reliance on arbitrageurs. Service providers may rely on persona providers to supply them with personas. Then, service providers interact anonymously with individuals. This moves the role of the guarantor of service providers from arbitrageurs (middle guarantors) to certifying authorities (backend guarantors).

There is an advantage of having backend guarantors, rather than having guarantors in the middle. The interest of users is to protect their privacy, whereas the interest of guarantors in the middle is reducing operational cost. Protecting user privacy is considered an operational cost. Thus, there is a conflict of interest between the guarantors and users. The interests of guarantors in the backend, such as persona providers, does not conflict with user interests. Therefore, having the guarantors at the back is better than having the guarantors in the middle.

There is another advantage of having certifying authorities compared to arbitrageurs. In the latter case, individuals must trust all arbitrageurs. This is due to the fact that service providers place offers at different arbitrageurs. However, in the first case, individuals have to trust very few certifying authorities. Since there are many arbitrageurs and few certifying authorities, individuals is better of trusting the few authorities than trusting the many arbitrageurs.

5.2 Background and Related Work

Anonymous double-auctions systems [72, 81] are perhaps the most related work to this chapter. In such a setting, buyers and sellers interact anonymously, and may

change roles from buyers to sellers and vice versa. Sellers sell their products through auctioneers. Buyers submit their bids to the auctioneers. Upon completing an auction, the auctioneer reveals the identity of the winner to finalize the transaction.

Although these efforts provide anonymity for bidders and sellers, bidders have to trust auctioneers to verify offers' attributes. Offers from different sellers cannot be combined anonymously; that is, double anonymous auction systems do not allow two sellers to prove to buyers that two offers are related, while being anonymous.

Onion routing [37] is a distributed anonymous communication protocol. Eavesdroppers on the network do not learn the IPs of the original sender or the final recipient of a message. When a message is sent from one node to another, the message is relayed from one *onion router* to another until it reaches the recipient node. These routers route messages unpredictability to achieve anonymity. Each onion router on the message path receives the message from that router's predecessor, decrypts one layer of the message to determine the next onion router, and passes the message to that router.

TOR networks [30] enhance and use onion routing to provide anonymous communication. TOR enables providers to publish hidden services. Users of TOR may access these hidden services, without knowing the IP addresses of the services. TOR assigns *.onion* domains to the services, by which they are accessed. Traffic from and to .onion domains are carried anonymously by TOR.

5.2.1 Related Work *vs.* Personas

Although the related work enables service providers to be anonymous, service providers are incapable of demonstrating the quality (attributes) of their services. For example,

a hotel may not be able to prove the presence of certain amenities in a room. Another problem is that the related work does not allow entities, whether individuals or service providers, to prove the existence of relations between them and other entities, in an anonymous fashion. This is needed in many scenarios. For example, to access a joint business account, two individuals must prove to their bank that they are co-owners.

Personas generalize anonymity to include service providers. This empowers service providers to interact with individuals anonymously, while proving service qualities to them. Personas also permit for anonymous interactions that involve a set of entities connected by a set of relationships.

5.3 Anonymity for Service Providers

By generalizing personas to allow service providers to have them as well as individuals, we overcome the shortcomings of the related work. Just like individuals, service providers can use personas to prove certain qualities to their customers. For example, a rating organization may provide the Hilton with a persona that attest the four-star class of this hotel. The Hilton can generate locked personas, which can be thought of as offers, and submit them to Priceline. Individuals surfing Priceline may verify the offer and the hotel class, without knowing the hotel identity. (Notice that Priceline is now providing only a meeting place and is not acting as a guarantor.)

This section presents an extension of the system presented in Chapter 3 and 4. The extension does not assume an asymmetry between the roles of individuals and SPs. Both individuals and SPs are entities that receive personas and use them to interact with other entities. The next two sections illustrate the extension in two e-commerce settings.

In the first setting, the extension is used to reduce the reliance on arbitrageurs. PPs provide personas to SPs and individuals. SPs generate locked personas that encode their offers and submit them to arbitrageurs. Individuals review these offers, and generate locked personas to purchase these offers. The role of arbitrageurs is limited to providing a meeting place.

In the second setting, the extension is used to remove the need for arbitrageurs. PPs provide personas to SPs and individuals. SPs generate locked personas that encode their offers and use a *.onion* domain per offer. Individuals review the offers, and generate locked personas to purchase these offers at these domains. The role of arbitrageurs is removed.

5.3.1 Reducing the Reliance on Arbitrageurs

In current settings, arbitrageurs take a slice of the profit in transactions by doing things that the transaction parties are unwilling or unable to do themselves. In the context of e-commerce, this usually means concealing the identity of a supplier while guaranteeing some of that supplier's properties. With the additional features of our persona system, a supplier can achieve both of these properties without the need for an arbitrageur, using a persona to conceal its identity, but able to provide guarantees. Of course, the role of an arbitrageur as guarantor has, to some extent, been moved to the persona provider who generates the personas.

What is missing when both customer and supplier use personas is a place for the transaction to be carried out. Arbitrageurs are used to provide such a place, but without the need to rely on them to guarantee the offers to individuals. This represents an advantage over the related work, since SPs are not charged by arbitrageurs

to attest the validity of SPs' offers to customers.

Figure 5.1 shows PPs providing both customers and SPs (a hotel and an airline company) with personas. The personas are generated as described in Chapter 4. The hotel sends its attributes (A_H) to the PP. The attributes of the hotel are, for example, the star rating and the presence of certain amenities. Similarly, the airline sends A_A to the PP. The individuals D_B and D_R sends their attributes A_B and A_R, respectively. PPs generate personas (P_H, P_A), along with corresponding secrets (S_H, S_A), that attest the validity of the SPs attributes. The same is done for the individuals D_B and D_R.

$$\{P_H,\ S_H\} = \mathit{Wrap}(A_H),\ \{P_A,\ S_A\} = \mathit{Wrap}(A_A)$$
$$\{P_B,\ S_B\} = \mathit{Wrap}(A_B),\ \{P_R,\ S_R\} = \mathit{Wrap}(A_R)$$

SPs generate locked personas (L_H, L_A) representing their offers and submit them to arbitrageurs, whereas customers generate locked personas (L_B, L_R) to purchase these offers.

$$L_H = \mathit{Show}(O_H,\ P_H,\ S_H),\ L_A = \mathit{Show}(O_A,\ P_A,\ S_A)$$
$$L_B = \mathit{Show}(O_B,\ P_B,\ S_B),\ L_R = \mathit{Show}(O_R,\ P_R,\ S_R)$$

O_H is a message encoding the hotel's offer, say "1 room, 1 queen bed, 1 night, 80$", and O_A encodes the airline's offer.

Customers may search arbitrageurs for offers, and verify these offers by verifying SPs' locked personas. SP's attributes are verified simultaneously.

$$\mathit{Verify}(L_H,\ A_H),\ \ \mathit{Verify}(L_A,\ A_A)$$

If an individual is interested in an offer at an arbitrageur, she generates a locked

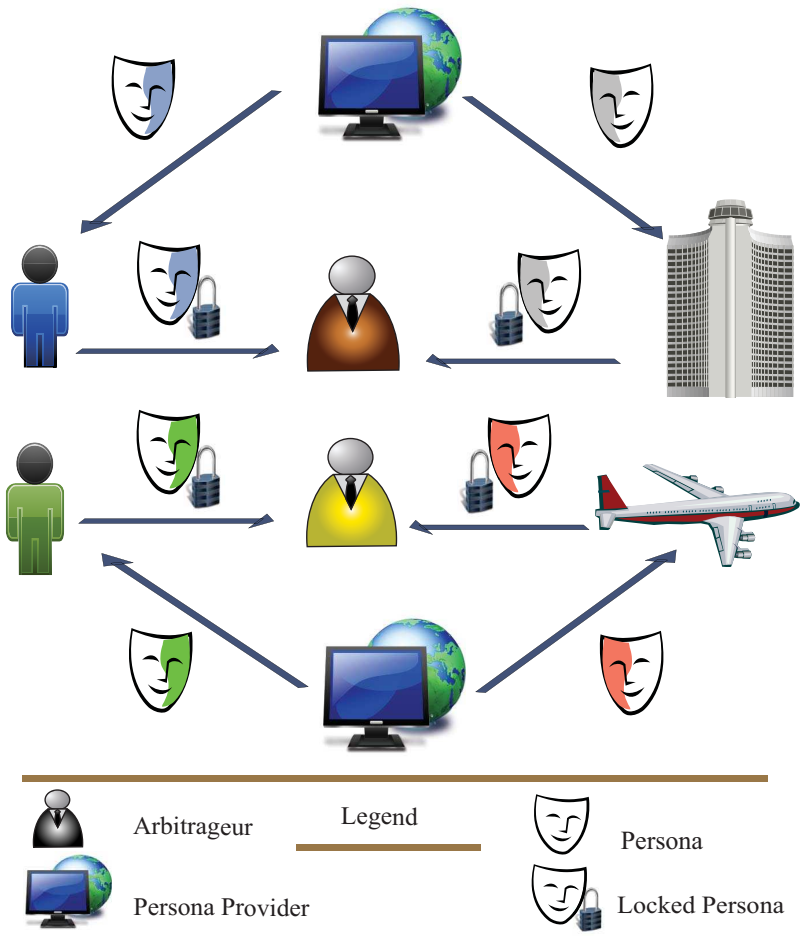

Figure 5.1: Reducing the reliance on arbitrageurs using personas

persona that shows her financial ability and sends it to that arbitrageurs. Arbitrageurs may verify the attributes of that individual by verifying customers' locked personas.

$$\mathit{Verify}(L_B,\ A_B),\ \ \mathit{Verify}(L_R,\ A_R)$$

Finally, the arbitrageur completes the transaction by revealing the SP's identity to the individual. This is achieved by de-anonymizing the SP's locked persona which represents that offer.

$$P_H = \mathit{Trace}(L_H),\ \ P_A = \mathit{Trace}(L_A)$$

Suppose that the hotel and the airline want to combine offers, say a 30% discount for flying with that airline and spending two nights at that hotel. The SPs ask a PP to generate two personas, P_H, P_A, that encode the relation between the offers generated by the personas. The SPs then generate two locked personas (L_H, L_A) that correspond to the combined offers. To verify the combined offers, customers use VerifyRelation to verify the locked personas.

$$\mathit{VerifyRelation}(L_H,\ L_A,\ \mathit{relation})$$

where *relation* is the encoding of the combined offers, say "30% discount on 2 nights and 1 ticket".

5.3.2 The End of Arbitrageurs?

Arbitrageurs are needed in to provide a place for SPs to publish their offers. However, a supplier can also provide this without compromising identity. For example, services hidden in TOR [30] networks could be utilized to post and claim offers. This does not violate the service provider's anonymity, since each offer can be posted and claimed on separate '*.onion*' site. For example, the Hilton could post an anonymous offer at

offerX.onion. Individuals may check the offer, buy it, and claim it at that address.

Figure 5.2 shows a setting similar to the one described in Figure 5.1. The difference is that the arbitrageurs are substituted with a TOR network. Personas are generated for SPs and individuals just like the previous section. Locked personas are also generated and verified as in previous section. The difference between the two settings is the mechanism of publishing SPs' offers and purchasing them by customers.

SPs publish their offers at *.onion* domains, rather than doing so at arbitrageurs. Each SP may host each offer at a separate domain. The domains of an SP are hosted at that SP's server. Individuals may use a search engine to search for offers. Individuals may verify the offers by verifying SPs' locked personas, as in the previous section.

To purchase an offer, an individual generates a locked persona and submits it to the domain hosting that offer. The domain verifies the customer's locked persona and completes the transaction by revealing the identity of the SP to that customer. Since each offer is published at a separate domain, the customer cannot use that offer to de-anonymize the SPs' other offers.

5.3.3 Challenges of Adopting Personas

It is challenging to convince service providers that their business model should change to adopt personas and minimize reliance on arbitrageurs. It is also in the interest of arbitrageurs that service providers rely on them to facilitate their transactions. Thus, arbitrageurs are discouraged from using personas. To overcome this challenge, individuals should pressure service providers to adopt personas and show a strong demand for minimizing reliance on arbitrageurs.

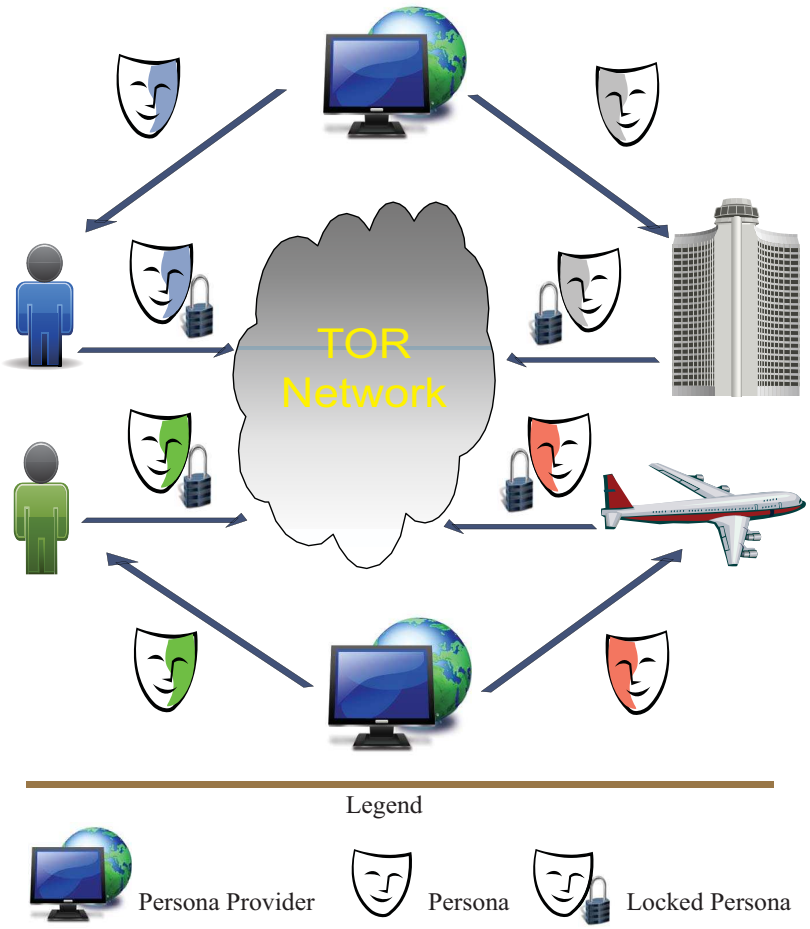

Figure 5.2: Removing the need for arbitrageurs using TOR Network

Individuals demand for a more privacy-preserving identity management has motivated many corporations to invest in privacy. Many service providers changed the way they conduct business to satisfy their customers expectation. For example, Liberty Alliance and WS-Federation are identity management initiatives by big corporations, such as Microsoft, IBM and HP. There is also another benefit for companies when they invest in new technologies. Such investment becomes an asset that raise these companies reputation.

5.4 Chapter Summary

There are an increasing number of scenarios that demand anonymity for service providers. In many B2C scenarios, service providers sell their surplus to individuals for lower prices through arbitrageurs. To avoid discouraging customers from buying directly from the service providers, at the regular price, those providers have to remain anonymous. Some providers wish to be anonymous for various reasons, including but not limited to, escaping denial of service attacks and avoiding censorship. This chapter applies personas in e-commerce settings. The objective is to allow service providers to remain anonymous, while reduce the reliance on arbitrageurs.

The motivation and our objectives are stated in Section 5.1. Section 5.2 describes the related work on anonymity for service providers. The differences of this work to the related work are addressed in this section. Section 5.3 presents an approach that permits service providers to participate in anonymous interactions with consumers, without requiring fully trusted arbitrageurs.

Chapter 6

Persona Applications: Access Control

Access to web services is regulated by access-control models, such as role-based access control (RBAC) [70]. These models specify who can access what service and when. Recent advances in web technologies demand more flexible access-control models that can handle the requirements of these technologies. For example, early access-control models have been designed for organizations with fixed set of users and services. These models do not scale well for new web technologies, like Semantic Web and Social Web, since users and services are expected to join and leave the system in an adhoc manner.

This chapter constructs an access-control model based on personas. The model protects the privacy of individuals, while enables the specification of access policies based on not only users' attributes, but also on the relations among these users. Section 6.1 states the objectives of this chapter. The related work is described in Section 6.2. Section 6.3 constructs an access-control model from the building blocks

of personas.

6.1 Objectives

Many access-control models, such as RBAC, are originally designed for closed systems where individuals, resources, and permissions are known *a priori*. This conflicts with the open nature of web technologies. In the new web, services are increasingly constructed from other services, each residing at a different service provider (SP). New services and resources are added to the system frequently and new permissions are created and assigned. Individuals use services and leave them in an *ad hoc* manner. Fixed permissions and access-control patterns do not fit well with dynamic environments like the new web.

There is also a privacy threat arising from the way services are constructed from other services. During the execution of a web service, the identity information of individuals is passed among these internal services. Supplying identity information to an SP implies that the individual is not sharing the information with that SP only, but also with a group of other SPs, whose existence may not be obvious, or even visible, to the individual.

To address the need for more flexible and privacy enhanced access-control models, certified attributes are used as the basis for access control [86]. Attribute-based access-control models support flexible and fine-grained access control as required by the new web technologies. The models also enhance the privacy by hiding individuals' identifying attributes, while allowing non-identifying attributes to be disclosed, possibly by enforcing some privacy policies.

Figure 6.1: An individual receiving and using a certificate

However, as described in the previous chapters, this approach is increasingly weakened by data-mining techniques. The attributes revealed may allow service providers to fuse identities and profile individuals.

This chapter applies personas as an access-control model. The objective is to construct an access-control system that suits the requirements of emerging web technologies. We illustrate the presented work with a scenario in the Semantic Web.

6.2 Background and Related Work

Given a set of individuals (subjects) and a set of resources (objects), an access-control system is responsible for deciding whether a subject has access permission to an object. Access control is an important component of security in software systems. Access decisions are based on the subjects' information available in their certificates, which they received from certificate authorities (Figure 6.1).

A large number of access-control models have been proposed. They differ in the type of information stored in certificates, and the way this information is used to access resources. This section discusses various access-control models. The discussion is limited to some of the well-known ones. We first summarize the access-control models. Then, the application of personas in access control is provided.

6.2.1 Identity-Based Access Control (IBAC)

In IBAC, access rights are assigned directly to individuals, Figure 6.2(a). Take for example access-control lists (ACL). Each resource is associated with a list of individuals who has access permissions over that resource. The list consists of identity-permission pairs, where each pair in the list specifies the identity of an individual and permission.

Another example is the capability-based access control [54]. Each individual is associated with a list of capabilities that the individual can perform. A capability is a reference to a resource with a set of permissions. Since capabilities consist of references to resources, they can be delegated from one individual to another. When an operating system successfully evaluates a request to open a file for reading, the operating system returns a reference to that file. The reference can be passed from one process to another (delegation).

One major problem with IBAC is that the management of identities and permissions becomes difficult as the number of individuals and resources grows. Since access is based on the identities of individuals, access control is coarse-grained and many policies cannot be implemented, for example, access control based on attributes. For example, even a simple attribute-based access such as "during working hours" is complicated by weekends, public holidays, and daylight-saving time changes.

6.2.2 Role-Based Access Control (RBAC)

RBAC [70] does not assign access permissions to individuals directly, Figure 6.2(b). RBAC assigns permissions to the roles of the individuals. The assignment of permissions has two parts: assigning permissions to roles and assigning roles to individuals. This is an advantage over IBAC. As long as the number of roles stays manageable,

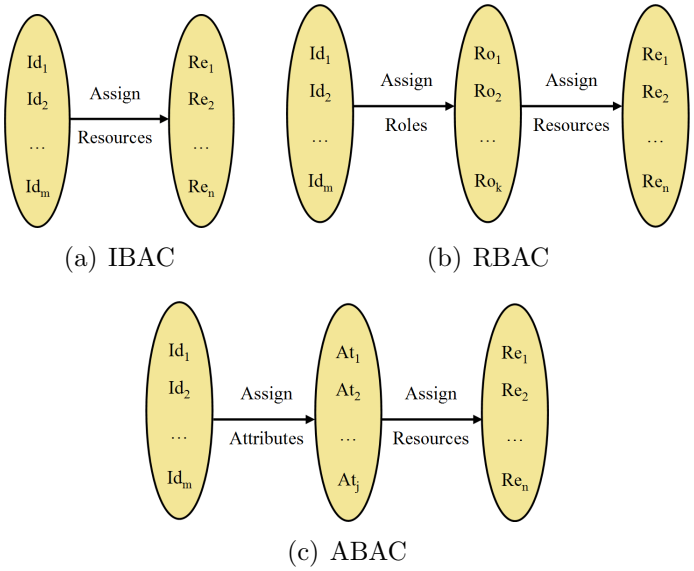

Figure 6.2: Access-Control Models

having large number of individuals and resources does not affect efficiency as in IBAC.

Several models for RBAC have been proposed, adding features such as role hierarchies [70], attributes and constraints [25], and contexts [53]. Role hierarchies facilitate the management of roles and help modeling actual roles in organizations. Adding the purpose of access to RBAC is presented by Byun *et al.* [20]. This model uses access purpose to protect sensitive data from unnecessary access. Temporal-RBAC [12] permits the enabling and disabling of roles based on temporal conditions, for example, activating a role based on a time period. Rule-Based RBAC [4] automatically assigns individuals to roles, based on their attributes. The model eliminates the need for manual assignment of roles.

RBAC models have several drawbacks:

1. RBAC does not suit the new web. Access to resources in new web technologies are based on users' attributes, rather than roles. Attributes allow for more

fine-grained access policies to be specified. For a mid-size company with fixed resources and fixed set of policies, RBAC works fine. However, this does not apply for the internet. Specifying access policies based on attributes is more convenient.

2. Roles in an organization grow quick because of new projects, new domains of business, and new activities. It may be difficult to update the role hierarchy in a timely way to reflect this.

3. Access cannot be specified based on relationships among individuals.

6.2.3 Attribute-Based Access Control (ABAC)

ABAC uses attributes as the basis for access-control decisions, Figure 6.2(c). By avoiding roles and using attributes in specifying access policies, ABAC does not suffer from RBAC's drawbacks. The work of Yuan [86] utilizes three components: individual attributes, resource attributes, and system attributes. Policies are used to specify access in terms of these components. This model supports fine-grained access-control policies. Policy formulation and enforcement for ABAC is also discussed.

An attribute-based access-control model is presented by Backes *et al.* [7]. The model allows an individual to prove the possession of certified attributes, rather than revealing the certified attributes themselves. The model, however, focuses on the scenario where an individual interacts with an organization. In many cases, interactions require the participation of a set of individuals, possibly connected by a set of relations. Protecting the privacy in this scenario is important, yet unaddressed.

6.2.4 Related Work *vs.* Personas

The certificates an individual uses to access web services are constructed based on the access-control paradigm. In IBAC, certificates contain identities. In RBAC, certificates contain roles. In ABAC, certificates contain attributes. There is a common privacy threat in IBAC, RBAC, and ABAC. In RBAC, individuals reveal attributes to assume roles to get access. In ABAC, individuals reveal attributes to satisfy policies to get access. Privacy policies may be used to regulate the flow of attributes and prevent the disclosure of identifying or sensitive attributes. As described in the introduction, this approach is increasingly weakened by data-mining techniques. The attributes revealed may allow service providers to fuse identities and profile individuals.

Personas can be used as certificates that prove the possession of attributes, rather than showing them. Although the ABAC model of Backes *et al.* [7] takes a similar approach, personas have an advantage of being capable of proving the existence of relations among individuals.

6.3 Guarantee-based Access Control

We present the guarantee-based access-control (GBAC) model. Our approach is similar to the attribute based access control. This allows access control to be fine-grained, avoids the burden of design and management of roles, and suits the open nature of new web technologies. However, instead of using individuals' attributes to construct certificates, as the case in ABAC, our access-control model uses guarantees about these attributes. These guarantees are used for access-control decisions. It resists threats to individuals' privacy, such as profiling. The model also permits access rights to be

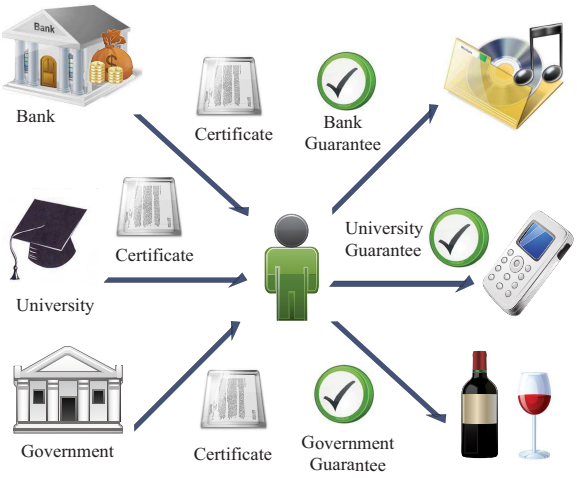

Figure 6.3: An individual receiving certificates and showing guarantees

based on a set of individuals in a particular structured relationship.

Figure 6.3 shows an authority providing an individual with a certificate. The individual uses that certificate to generate a guarantee to be used to access a resource. In GBAC, certificates are implemented by personas, while guarantees are implemented by locked personas. Suppose that an individual receives three certificates from her government, bank, and university, as shown in Figure 6.3. Figure 6.4 shows part of the access policies at the wine, music, and phone stores. Each store specifies the required access policies based on guarantees and enforces them using a policy enforcement framework as shown in Figure 6.5. The specification of policies may take other factors into consideration: the service status, the context, and temporal conditions.

```
/* Authorization policy at the wine store                                    */
accepted_age_guarantee_provider = { government, ... }
if Provider(guarantee) ∈ accepted_age_guarantee_provider &&
Guarantees(guarantee) = legal_drinking_age then
 |  proceed to checkout
else
 |  abort transaction
end

/* Authorization policy at the music store                                   */
accepted_credit_guarantee_provider = { BMO, HSBC, ... }
if Provider(guarantee) ∈ accepted_credit_guarantee_provider &&
Guarantees(guarantee) = available_credit then
 |  proceed to delivery
else
 |  abort transaction
end

/* Authorization policy at the phone store                                   */
accepted_student_guarantee_provider = { ISIC, ... }
if Provider(guarantee) ∈ accepted_student_guarantee_provider &&
Guarantees(guarantee) = student then
 |  proceed to student offers
else
 |  abort transaction
end
```

Figure 6.4: Part of the authorization policies at stores

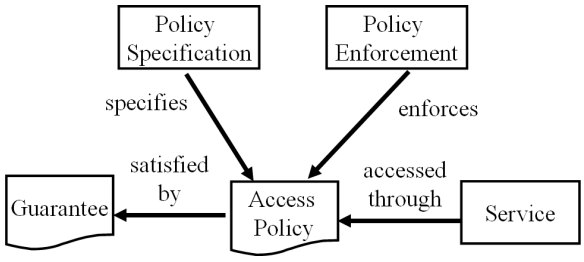

Figure 6.5: GBAC from a service provider's perspective

6.3.1 Using Personas to Implement Guarantees

The following describes how personas are used to implement guarantees. PPs play the role of authorities. Assume that an individual requests an SP to provide access to a resource. The SP responds by specifying which attributes the individual needs to send to SP.

If the individual does not have the required persona, the individual sends her attributes (A) to the PP. The PP generates a persona (P), along with corresponding secret (S), that attest the validity of the individual's attributes. The individual uses P to generate a locked persona (L), and submits it to an SP. The SP verifies L to check whether the individual is entitled to the claimed attributes.

$$\begin{aligned} \{P,\ S\} &= \mathit{Wrap}(A), && \textit{at PP} \\ L &= \mathit{Show}(P,\ S), && \textit{at individual} \\ \{\mathit{true},\ \mathit{false}\} &= \mathit{Verify}(L), && \textit{at SP} \end{aligned}$$

The SP loads the policy Y associated with the requested resource R. The SP then evaluates the policy given the claimed attributes. If the policy evaluates to true, the SP allows the individual to access the service R. *GetPolicy* returns a policy given a resource R, while *EvaluatePolicy* determines whether a policy Y is satisfied by a set of attributes A.

$$\begin{aligned} Y &= \mathit{GetPolicy}(R) \\ \{\mathit{true},\ \mathit{false}\} &= \mathit{EvaluatePolicy}(Y,\ A) \end{aligned}$$

6.3.2 Access Policy Specification and Enforcement

The management of access policies differs from one SP to another. The generation, verification, and tracing of guarantees are independent from policy management. The *GetPolicy* and *EvaluatePolicy* operations can be implemented independently from guarantees. Therefore, access-policy specification and enforcement in GBAC are not restricted to a specific framework. The example provided in the next section discusses a candidate policy framework.

6.3.3 GBAC and the Semantic Web

To illustrate GBAC, the Semantic Web [11] is used as an example. The Semantic Web is an extension of the World Wide Web. The extension works by adding semantics to the content and resources of the web. Web languages, like HTML, make the content of web pages readable for humans. Semantic-Web languages, like OWL [60], make the same content readable for machines, allowing machines to not only process the content of a web page for rendering purposes, but also to understand and reason about the meaning of the content. This is achieved using web ontologies. Web Ontologies are at the heart of the Semantic Web. Service discovery, invocation, and composition are all based on the availability of machine-readable web ontologies. An ontology is a description of a set of things (concepts) and the relations among these things. Normally, an ontology is an attempt to formally describe a domain.

For example, an ontology that describes vegetables may list the characteristics of each group / type of vegetables, but not the specific values of these characteristics for each species of vegetables. The values for a specific specie are described in a separate document, based on that ontology.

```
<Ontology>
  <Class id = "Authority">  <Class id = "SP">  <Class id = "PublicParameters">
  <Class id = "Certificate">  <Class id = "Person">  <Class id = "Guarantee">
  <Class id = "Resource">  <Class id = "Policy">  <Class id = "DA">

  <ObjectProperty id = "issuedBy">
    <domain resource = "Certificate">
    <range resource = "PP">
  </ObjectProperty>

  <ObjectProperty id = "issuedTo">
    <domain resource = "Certificate">
    <range resource = "Person">
  </ObjectProperty>

  <ObjectProperty id = "generatedBy">
    <domain resource = "Guarantee">
    <range resource = "Certificate">
  </ObjectProperty>

  <ObjectProperty id = "hasACertificate">
    <domain resource = "Person">
    <range resource = "Certificate">
  </ObjectProperty>

  <ObjectProperty id = "publishedBy">
    <domain resource = "PublicParameters">
    <range resource = "Authority">
  </ObjectProperty>

  <ObjectProperty id = "hasAResource">
    <domain resource = "SP">
    <range resource = "Resource">
  </ObjectProperty>

  <ObjectProperty id = "satisfiedBy">
    <domain resource = "Policy">
    <range resource = "Guarantee">
  </ObjectProperty>

  <ObjectProperty id = "protectedBy">
    <domain resource = "Resource">
    <range resource = "Policy">
  </ObjectProperty>

</Ontology>
```

Figure 6.6: An ontology describing the concepts in the GBAC model

Figure 6.6 shows an ontology, in OWL, which is used by the entities to understand and process guarantees. The ontology describes the concepts of a person, authority, SP, DA, certificate, guarantee, resource, and policy. A certificate is issued, by an authority, to a person. An authority publishes a set of public parameters. A guarantee is generated by a certificate. Persons use these parameters, along with their certificates, to generate guarantees. SPs use these parameters to verify guarantees. An SP has a set of resources. A resource is protected by a policy, which can be satisfied by a guarantee.

Each entity publishes an RDF file describing that entity, based on the ontology. For example, each PP publishes an RDF file that describes that PP's properties. The RDF file contains the web-address of the public parameters required to use and verify personas that the PP generates. Each SP publishes a RDF file describing the web-address of its resources, and the access policies associated with these resources.

Figure 6.7 shows the architecture of GBAC in the context of the Semantic Web. When an individual wants to acquire a guarantee, the individual-side client uses an ontology to discover the appropriate authority. Once the individual chooses an authority, the client uses an ontology to communicate with that authority to get the required certificate. The SP also uses ontologies to understand the meaning of guarantees.

The client uses an ontology to search for SPs, based on the individual preferences. Once an SP is chosen, the client negotiates with the SP the needed guarantees. The client uses her certificate to generate guarantees and sends them to the SP.

The handler, at the SP, relays the access request to a policy engine, which reasons over the access policies and the provided guarantees. The policy engine replies with a

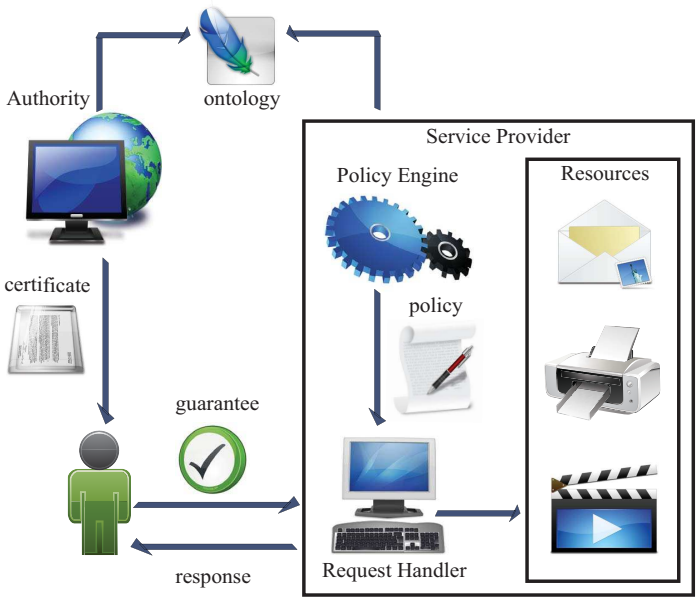

Figure 6.7: An architecture of GBAC in a Semantic-Web settings

decision whether to permit access or not. REI [50] is an engine for specification and reasoning over access policies for resources on the Semantic Web. REI specifies access policies using OWL-S constructs. OWL-S is an extension of OWL for web services. Finally, the handler either allows the individual to use the service or denies her.

6.4 Chapter Summary

Access-control models regulate access to web services. The models specify what services an individual may access, under which circumstances. Traditional access-control models, like RBAC, do not suit the requirements of new web technologies. For example, early access-control models have been designed for organizations with fixed set of users and services. These models do not scale well for new web technologies,

like Semantic Web and Social Web, since users and services are expected to join and leave the system in an adhoc manner.

The objectives of this chapter are stated in Section 6.1. Section 6.2 surveys the related work. In Section 6.3, the building blocks of personas are used to construct an access control.

Chapter 7

Persona Applications: Reputation Management

Trust is a vital aspect in our daily life, especially on the web. It allows individuals to participate in interactions and take important decisions, without the necessity of having previous experience with those whom they interact with. For example, in peer-to-peer (P2P) networks, the more trust a peer has in another peer, the more encouraged the first peer is in interacting with the latter one. Another benefit of trust management is in information dissemination. Trust enables individuals to release sensitive information to trusted entities only, which preserves their privacy. One way to quantify trust is based on reputation. The reputation-based approach for establishing trust among entities is a well-researched field [28, 51, 62, 69, 75]. The feedback of individuals regarding their interactions with others is an important criterion when calculating reputations. The more feedback the individuals provide, the more accurate the reputation values that are computed.

This chapter constructs a reputation-management system for P2P networks. The

system is built on top of our secure interaction system, described in Chapter 3. The objectives of the presented system are presented in Section 7.1. Section 7.2 surveys the related work and shows how it differs from our system. Section 7.3 extends our persona system to provide the support for reputation-management functionality. The design issues of the system are discussed Section 7.4.

7.1 Objectives

There is a problem in reputation-based trust, from an anonymity perspective. To accurately compute the reputation of an entity, individuals should provide feedback regarding their interactions with the entity, possibly, accompanied with transcripts. Alternatively, the individuals must permit the entity to provide the interactions' transcripts to the reputation system. This is problematic, since in both cases, it helps the system to profile the individuals. In fact, the more an individual participate in the feedback process, the more her actions are susceptible to profiling.

There is another problem that limits the practicality of current work on reputation management. The work fails to reward products and services that did not receive ratings, due to the low participation of the individuals using them. For example, the rating process may not be easy to use. In some cases, individuals may not even participate at all. Individuals may get discouraged from rating certain products for sensitivity, religious, and political reasons.

Anonymous reputation-management (ARM) systems [29, 62, 75] permit the anonymous rating of products and services; yet they fail to address the following.

- All ratings by one peer are linkable to each other, which leads to profiling and re-identification of that peer.

- A service may get a reputation score lower than others, due to the significant gap between the number of peers who used the service, and those who rated the service. ARM systems do not differentiate between such a service and those services that peers did not use in the first place. This represents a disadvantage for services which do not get frequently rated. Services become pressured to put more effort in convincing peers to rate them, than in enhancing the services' quality.

- ARM systems suffer from the effects of Sybil attacks [31]. In Sybil attacks, the attacker creates multiple accounts to be used to submit good/bad reputation values to gain some advantage. The systems either do not address Sybil attacks, or employ techniques that have negative effects on the usability and performance of the system.

This chapter presents a system that facilitates reputation management, while avoiding the described problems. The system is based on empowering individuals to securely use web services, without enabling service providers to profile those individuals. Services gain reputation even if individuals who use the services neglected to rate them. The presented approach is decentralized; that is, no single authority is needed to compute and manage the reputations. This makes the approach suitable for distributed systems, like P2P networks. Service providers may also specify constraints on the rate that a persona can use a service in a specific time interval. This feature is used to limit the effects of Sybil attacks on the system.

The next section describes reputation-based trust, mainly in P2P networks, and explains why the related work, anonymous reputation management [29, 62, 75], fails to protect individuals' privacy.

7.2 Background and Related Work

Grandison *et al.* [40] define the trust in an entity as the belief in the competence of that entity to perform a task in a dependable and reliable manner, in a specific context. Trust plays a major role in reasoning about the authenticity and quality of information. The work on trust is usually categorized into two main paradigms, policy-based [10, 49, 85] and reputation-based. In policy-based, an entity establishes trust in another by examining the credentials the former entity possesses. Policies are used to determine the level of trust in that entity. For example, an entity A may trust an entity B only if it possesses certificates Y and Z. In reputation-based, the history of the interactions that an entity had with others are used to evaluate the entity's trustworthiness. For example, one peer in a P2P network may trust another peer if the latter has been recommended by P number of peers [6].

A careful look, however, reveals that the two paradigms are the same. In fact, one can treat both paradigms as distributed reputation-based trust. This is because trust evaluation in both paradigms is distributed among entities. Trustworthiness is determined based on the reputation of the entities which perform the evaluation. An entity A trusts an entity B due to the recommendations of C and D (or credentials provided by C and D). The level of trust from A to B is based on the reputation of C and D themselves. It is also clear how trust is recursive; A trusts B since it trusts the issuer of B's credential and so on.

7.2.1 Reputation-based Trust in P2P Networks

There are many metrics used to compute the reputation of an entity, for example, the feedback from other entities, the number of successful interactions, and seniority.

Note that the metrics differ from one application to another. In applications where privacy is essential, one finds metrics measuring the ability of entities to preserve the privacy of others.

The EigenTrust algorithm [51] computes a reputation score for peers in a P2P network. The reputation is computed based on the PageRank [19] algorithm. In the web of trust [36] approach, each entity maintains the reputation information about its neighbours. To determine the level of trust an entity A should have in an entity B, A uses a trust metric which specifies how A should traverse the different paths to B, and how to aggregate the reputation values. Rezgui *et al.* [69] use a reputation-based approach to protect the privacy of individuals. The approach evaluates the trust of web services based on several metrics. Those services with low reputation score are monitored more frequently than those with higher reputation.

One problem of the described reputation management systems is that they do not allow individuals to participate in the system anonymously. This discourages individuals from participation, especially in the context of evaluating the reputation of services and products that are sensitive, religious, or political in nature. Anonymous reputation management (ARM) [29, 62, 75] tackles this problem by empowering individuals to participate anonymously. The following describes some efforts in building ARM for P2P networks.

In TrustMe [75], the reputation information of each peer in a P2P network is collected, stored, and updated by a randomly assigned set of peers, called the Trust-Holding Agent (THA) peers. If peer i wishes to submit a reputation value for peer j, peer i signs this value, encrypts it with the keys of the THA responsible for peer j, and broadcasts it over the network. The THA peers of peer j can read the value and

update its reputation.

SuperTrust [29] is another ARM system. Just like TrustMe, SuperTrust assigns the management of the reputation of a peer to a set of super peers. A unique feature of SuperTrust is the use of a homomorphic encryption function. The function allows a super peer to aggregate the encrypted reputation values of its assigned peers, without decrypting the values. Thus, the reputation values that a peer i submits to super peers remain secret.

Muler *et al.* [62] present an ARM that provides anonymity for peers, while protecting against Sybil attacks.

While these ARM attempts provide anonymity, there is a problem that remains unresolved. Peers sign reputation values with their public keys, and then submit these values to the THA and super peers, in TrustMe and SuperTrust, respectively. The feedback records of a peer, although anonymous, are still linkable to each other. This is problematic from a privacy perspective. The ability to link individuals' actions implies the ability to build anonymous profiles, which can be linked back to individuals.

7.2.2 Related Work *vs.* Personas

As described in the problem statement of this thesis (see Section 1.2), data-mining techniques enable one to link partial identities together. This suggest that anonymity is not enough. To protect an individual's privacy, the reputation values submitted by that individual should be unlinkable to each other. The next section presents a new approach that handles the linkability problem, which exists in the current work on ARM.

7.3 Anonymous Reputation Management for P2P Networks

A system that supports personas can be tweaked for reputation management. This section describes the usage of personas to construct a reputation-management system for P2P networks.

7.3.1 Personas in P2P Networks

Personas can be used in P2P networks to allow peers to interact with each other, both securely and anonymously. Recall from Section 3.1 that there are four entities manage personas: individuals, persona providers (PPs), service providers (SPs), and de-anonymization authorities (DAs). To apply personas in P2P networks, the four entities are mapped to the entities of a typical P2P network. Individuals are the peers in a P2P network. PPs are the servers responsible for authenticating peers, upon joining the network, and providing them with credentials. SPs are special peers who provide services to peers, based on their credentials. DAs are the servers that are responsible for tracking peers abusing the network policies.

7.3.2 The Persona Approach for Reputation Management

Recall that a locked persona and a signed response an individual gets from an SP are a proof of interaction between that individual and that SP, in the form of requesting and providing a service. An individual can submit a locked persona, along with a reputation value measuring the satisfaction by the service, to a reputation-management system. The system updates the reputation score of the SP's service and the SP itself.

Thanks to the unlinkability property of locked personas, individuals may submit all their locked personas to the reputation-management component, without the fear of being profiled. This has two significant implications. First, the reputation scores an individual submits are anonymous. Second, the reputation scores are unlinkable to each other. While current anonymous reputation management (ARM) systems achieve the first, they fail to achieve the second. Modeling reputation using personas is, therefore, more privacy preserving than previous ARM systems.

The unlinkability of reputation scores may enable some individuals to launch Sybil attacks. The presented system prevents such attacks by disallowing an individual from submitting more than one reputation score for an SP, in a given time interval. This is achieved by the constrained-interactions property of personas. Section 7.3.3 presents the approach in more detail.

Another advantage that our approach has is the following. An SP needs not to wait for individuals to submit their reputation scores for that SP. The SP may simply submit the locked personas it receives from individuals to the reputation-management component. The component can utilize the locked personas to give the SP partial reputation reflecting the fact that the SP is trustworthy enough to motivate individuals to request that SP's service. Recall that the well-known PageRank [19] algorithm works by counting the number of links to a page, rather than whether the links have good or bad connotation. EigenTrust [51] is a well-known reputation management system that utilizes PageRank.

Now we turn to the method of querying, submitting, and updating reputation scores of SPs. We introduce reputation-management components (RMC) to the persona system. These components respond to individuals when they query for an SP

reputation, collect reputation scores from individuals, and update SPs reputation accordingly. Note that our approach focuses on facilitating the functions of an ARM system. Thus, our approach should work well with any reputation-aggregation metrics. We also do not mandate a specific method for choosing RMC locations in the network, and the assignment of SPs to RMCs. Instead, this is left to system administrators.

- **Query for a reputation**. An individual queries RMCs for an SP's reputation.
- **Submit a reputation**. An individual submits a reputation message to an RMC.
- **Update a reputation**. An RMC updates the reputation score of an SP, based on a reputation algorithm, *e.g.*, EigenTrust [51].

Suppose that an individual D wishes to submit a reputation score for an SP to an RMC. D prepares a reputation message as:

$$Reputation = \{score, SP, locked\ persona, SP\ response\} \tag{7.1}$$

where *SP* is the identity of the SP, and *SP response* is the signed message that D receives from *SP* as a response for a service request. D then generates a new locked persona to prove that D is the individual who used the service and reported the score. Note that D makes the new locked persona linkable to the original locked persona. D sends the tuple (reputation message and the new locked persona) to an RMC. We refer to the tuple as signed reputation. Finally, the RMC verifies that the locked persona in the reputation message and the locked persona, generated based on the message, are valid and linkable to each other. In other words, both locked personas are generated by D.

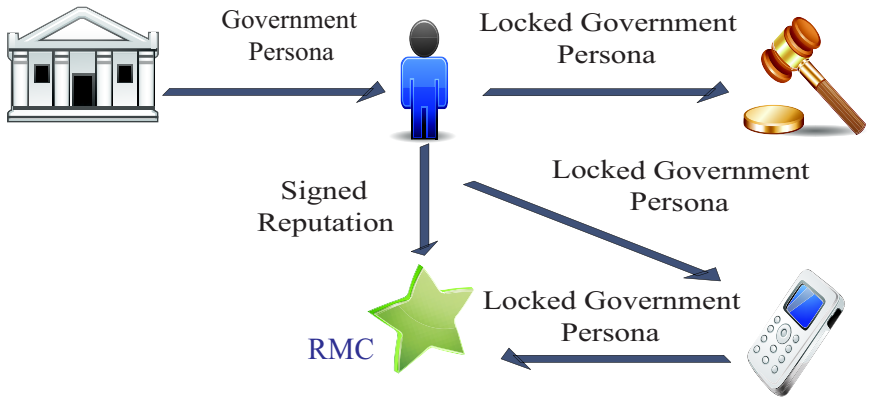

Figure 7.1: The persona-based reputation approach

Figure 7.1 shows an individual using a government persona to generate locked personas, to interact with an auction site and a phone store. There are two scenarios present in the figure. In the first, the individual submits a reputation for the auction site to the RMC. In the second, the phone store submits a locked persona to the RMC.

The second scenario is needed to allow SPs to gain reputation, even if their customers did not submit reputation scores for these SPs. The more locked personas are used at a service provider (SP), the more reputation that SP should have. The RMC assigns reputation values for the SP based on the rate of the transactions made by their individuals at that SP. Of course, the RMC can detect the case where both the individual and SP submit reputation and locked persona for the same interaction. This is because the locked persona is the same in both cases.

Recall that locked personas encode the date of the interaction between individuals and SPs. This can be used by RMCs to check the frequency an SP's service or product is used by individuals. RMCs may use this frequency as a reputation metric, such

that when the frequency consistently increases or decreases, the reputation is also updated accordingly.

7.3.3 Preventing Sybil Attacks

To prevent Sybil attacks, RMCs do not accept reputation scores submitted by individuals unless they are accompanied with tickets generated by persona providers. Recall that tickets contain information about the number of times individuals used services. An RMC can use this information to disallow an individual from submitting more than one reputation score for a specific service, in a given time interval.

If an individual wants to rate an SP, the RMC requests the individual to submit a reputation message, as described in Section 7.3.2. The individual also contacts her PP to receive a ticket to rate the SP, then sends the ticket to an RMC. The RMC follows the steps of Section 7.4.2 to verify the ticket.

7.3.4 Preprocessing Reputation Messages

RMCs preprocess reputation scores before using them to update SPs' reputations. The preprocessing step is needed for practical issues, for example, avoiding malicious attacks and smoothing the effect of outliers. Such step exists in other reputation systems. For example, the trust values in EigenTrust [51] are normalized to the interval [0,1]. This disallows individuals from providing arbitrary high or low scores. Another useful preprocessing step is truncation to get rid of outliers, that is, removing a fixed percentage of the reputation scores from both end of the spectrum. For example, one may truncate 5% of the scores from both sides.

7.3.5 Reputation for Individuals

Personas may encode reputation for individuals. Assigning reputation values for individuals is important. It helps service providers to determine the trustworthiness of individuals; and it allows RMCs to give more weight for the reputation scores submitted by reputable individuals. Reputation should be assigned in a privacy-preserving manner. For example, a persona provider assigns a reputation level for a person based on money spent, and time passed since registration.

A persona provider may encode reputation levels as attributes in personas. This is similar to information assurance in identity-management systems, where an identity provider complements identity assertions with assurance values. These values represent the level of certainty that the identity provider has with respect to the assertions. Alternatively, the persona provider may use different sets of parameters to generate personas, where each set corresponds to a level of reputation. An individual generates a locked persona and claims a level of reputation. To verify the individual's locked persona, a service provider uses the persona provider's parameters corresponding to the reputation claimed.

7.4 System Design

This section presents the detailed design of the reputation management system. The security of the system is discussed in Appendix A.

7.4.1 Reputation-Management Operations

Reputation management requires five operations: submit reputation, verify reputation, update reputation, generate ticket, and verify ticket. Individuals submit reputation scores, along with tickets generated by PPs, whereas RMCs update reputation scores, after verifying the tickets. Tickets generation and verification are described in Section 4.3.5.

`SubmitReputation`:

locked persona $\leftarrow$ *SubmitReputation*(*persona*, *secret*, *reputation message*)

Suppose that an individual D wishes to rate an SP. The operation takes from D a reputation score sc and a proof of interaction with SP. The proof consists of a locked persona lp from D's side, and an SP's signed response sr from SP's side. A reputation message m is constructed as $m = \{sc, SP, lp, sr\}$. `SubmitReputation` then executes $\texttt{Sign}_e(m)$ to generate a locked persona $\bar{lp}$, where $\bar{lp} = \{\sigma, m\}$. Further, $\bar{lp}$ and lp must be linkable. The individual sends $\bar{lp}$ to the RMC as a reputation score for SP, by D. The individual should also submit a ticket to the RMC, see Section 4.3.5.

`VerifyReputation`: *boolean* $\leftarrow$ *VerifyReputation*(*locked persona*)

`VerifyReputation` receives a locked persona $\bar{lp}$, which includes a reputation message m. The RMC extracts SP, sr, lp, sc from m and uses $\texttt{Verify}_e$ to check whether lp and $\bar{lp}$ are valid locked personas and linkable to each other. The RMC also checks whether sr is a valid SP response. If all tests are passed, the RMC executes `UpdateReputation`.

`UpdateReputation`. The RMC updates the reputation score of SP, based on sc. Choosing which score aggregation algorithm to use and updating SP's reputation are left for system administrators.

7.4.2 Ticket Application: Prevention of Sybil Attacks

The individual prepares a ticket request m_I, as in Equation 4.22, where SP is the SP that the individual wishes to rate. The individual sends m_I to her PP. The PP generates a ticket t, as in Equation 4.23, and sends it back to the individual. The individual sends m_I and t to the RMC. The RMC uses Equation 4.24 to verify t. Recall that t includes n, which represents the number of times that the individual has requested a ticket for the same SP, in a given time interval. Finally, the RMC allows the individual to rate the SP only if $n = 1$, that is, this is the first time a ticket is generated on behalf of the individual, for the specified SP.

Note that the RMC should make sure that the locked persona in the ticket is the same locked persona that the individual submits in the reputation message.

The described algorithm limits the ability of an individual to rate an SP to once per time interval, while it prevents the PP from knowing which SPs are being rated.

7.5 Chapter Summary

An anonymous reputation management system for P2P networks is presented in this chapter. The features provided by personas are utilized to build the required functions, like submitting and verifying reputation messages. Section 7.1 shows the objectives of the presented system. The background and related work to our system are described in Section 7.2. Section 7.3 shows our approach to reputation management and how we achieve it using personas. The design issues of the system are addressed in Section 7.4. The prevention of Sybil attacks is also discussed in this section.

Chapter 8

Persona Applications: Cloud Computing

In cloud computing, the Internet is viewed as a web of resources. Cloud providers allow customers to access their resources, on demand. For example, researchers from one university may perform experiments using the labs of another university. Providers may also provide the software and infrastructure to businesses. For example, Amazon's storage (S3) and computing (EC2) services [5] provide businesses with on demand storage and computing power. Those providers require identity-management systems (IMS) to manage the identities of their customers and regulate customer access.

For example, business A may outsource the management of their database to a cloud provider B. A needs to tell B which employees has access to which tables. When A's employees require access to the database, B needs to authenticate them; therefore, B needs to manage the identities and / or certificates of the employees. The expected market size of cloud computing, which is estimated by Merrill Lynch

to surpass $100 billion [43], is a real motivation for IMSs.

This chapter applies personas in cloud computing to construct an IMS. The architecture of the IMS makes it more suitable for cloud-computing environments. Section 8.1 introduces cloud computing and its benefits. The section shows current work on identity management for cloud computing. Our approach for identity management for cloud computing is described in Section 8.2. The section also shows the advantages the presented IMS has over related work. Section 8.3 summarizes the chapter.

8.1 Background and Related Work

Cloud computing is a new paradigm where software, platforms, and infrastructure are treated as virtualized units that are accessed by consumers [33]. Cloud services are provided on demand and governed by service level agreements between providers and customers. Customers can be businesses, education and research institutions, governments, as well as individuals. A cloud provider normally has a large number of clusters, supercomputers, and servers.

Cloud computing services can be categorized into: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS); refer to Figure 8.1. In SaaS, providers license software for customers. For example, Salesforce.com provides its customers with a license to use sales data-analysis software over the net. In PaaS, providers offer environments and development tools for their customers to develop and run their applications, but customers are limited to the Application Programming Interface (API) provided by the provider. For example, the Google App Engine [39] allows customers to develop web services that are based on the Google API. In IaaS, customers are provided with the required software and hardware to

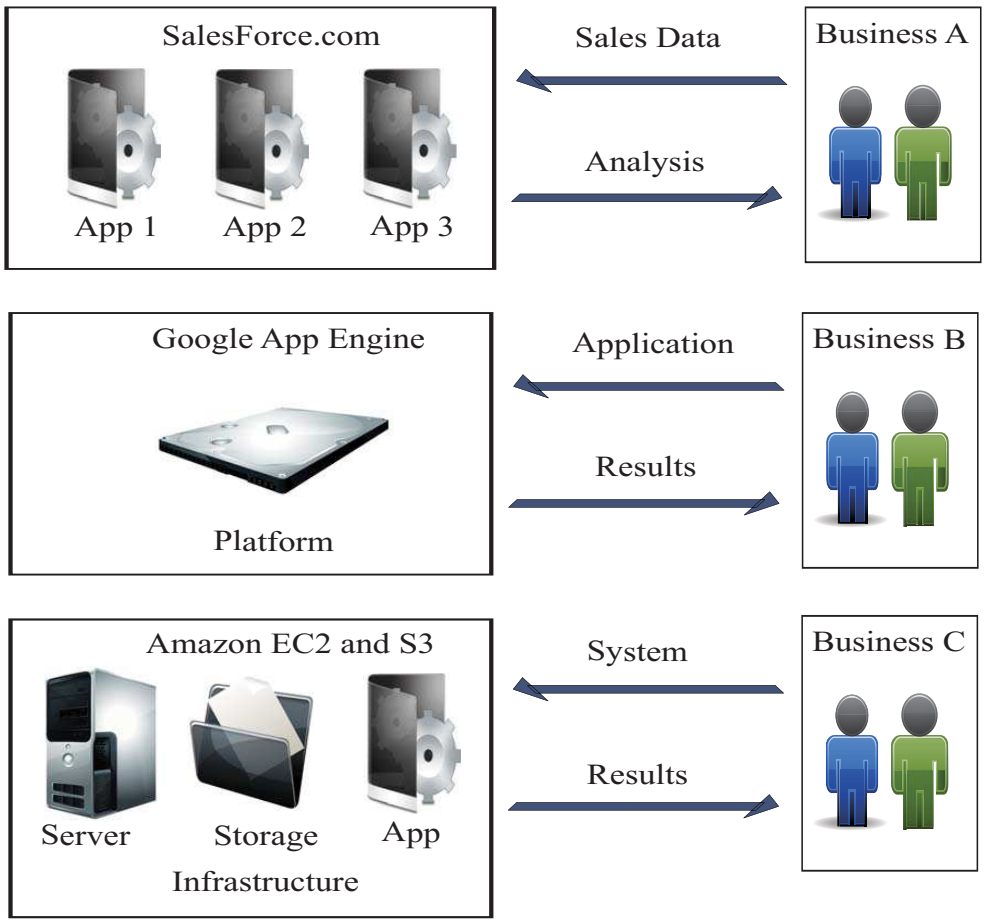

Figure 8.1: Examples of cloud computing

develop and run their applications. For example, Amazon S3 and EC2 services enable customers to store data and run applications on Amazon servers.

There are many benefits for cloud computing. First, businesses need not purchase software licenses and hardware that are not needed all the time. Instead, businesses may rent the required software or infrastructure, on demand. This reduces not only the cost of implementing applications, but also the operational cost of these applications. Second, since cloud providers invest in software and hardware resources, they

are able to implement and provide for more scalable and reliable solutions. Third, cloud computing also increases availability in the sense that it allows customers to access the required applications over the Internet.

8.1.1 Identity Management for Cloud Computing

There are many security and privacy issues that need to be addressed. For example, are providers' infrastructures secure enough? Will providers make sure that businesses' sensitive information and the identity information of their customers are kept private? What if a business resides in one country and its cloud computing provider resides in another country, where each country has its own surveillance and privacy laws? These issues motivate the use of various techniques to ensure the security and privacy of applications and data in the cloud. The following paragraphs discuss recent research on protecting privacy in cloud computing settings.

One of the tools that minimizes privacy risks are IMSs. There are some IMSs that may be applied in cloud computing, for example, federated IMSs. Another approach is the use of privacy policies to specify and enforce privacy regulations and laws. Creese *et al.* [26] present a capability maturity model to assess the security and privacy of a cloud provider. Design patterns are used to construct controls that help mitigate security and privacy risks. Service level agreements are used to ensure that providers implement the controls agreed on with customers.

Bertino *et al.* [13] present an IMS that enhances privacy and interoperability. Zero-knowledge proof protocols are used to convince verifiers that customers possess certain attributes, without disclosing the actual values of these attributes. Instead, customers may prove to verifiers statements about these attributes, *e.g.,* age > 18.

Interoperability is enhanced using a Semantic-Web language that defines the meaning of attributes. This allows for matching attributes that may differ across providers, but refer to the same concept.

Yan *et al.* [84] use identity-based cryptography along with federated identity management to address the case where each cloud contains multiple clouds. The proposed scheme simplifies the authentication of customers by achieving single sign-on.

Cáceres *et al.* [21] present Virtual Individual Servers (VISs). A virtual sever is a server that can be moved from one physical server to another. Each individual has a VIS that manages his/her data and can be used to specify the policies that regulate the flow of sensitive data. These VISs are hosted at cloud computing providers. Individuals send their data to the VIS with their mobile phones.

8.1.2 Cloud-based Security

The related work mentioned earlier applies existing techniques to secure and enhance the privacy of customers' data, including their identity information. A common factor among the related work is the limited utilization of the computation and storage resources that cloud computing providers offer, for the purpose of enhancing security and performance. Utilizing the computation and storage power of cloud computing, while designing security solutions, is called cloud-based security [63]. In cloud-based security, the software developer no longer has to worry about computation and storage limitations of the client machines. Major processing is done at a provider's cloud. Gartner Inc, a leading firm in information technology research, expects that cloud-based security will triple in many segments by 2013 [35].

Muttik *et al.* [63] design anti-virus software that keeps virus and malware signatures at designated clouds, while client machines communicate with these clouds to detect viruses. This minimizes the reliance on the computation and storage abilities of the client, as well as the time needed to deliver the updates of new virus signatures to clients. One disadvantage is the delay that may occur due to querying the cloud.

The next section presents a cloud-based implementation of personas that enhances the privacy of individuals.

8.2 Cloud-based Implementation of Personas

We use a cloud-based security approach to design an IMS for applications that run on the clouds. The IMS uses personas, as in previous chapters. The difference is in the use of new entities called *virtual persona servers*. Each one of these servers stores the personas of an individual. Individuals upload these servers to cloud providers, that provide a service of hosting such servers. An individual may also host a VPS at her machine. When an individual wishes to access a resource at a service provider A, that individual contacts her VPS to use her persona. The VPS generates a locked persona and uses it to interact with A. See Figure 8.2 for an illustration.

A VPS has the following life cycle.

- An individual collects personas from her persona providers, and instantiates a VPS to manage the collected personas.
- The individual uploads the VPS to a cloud provider.
- The VPS use the personas on the individual's behalf, at service providers. Service providers can be other cloud providers which provide services.

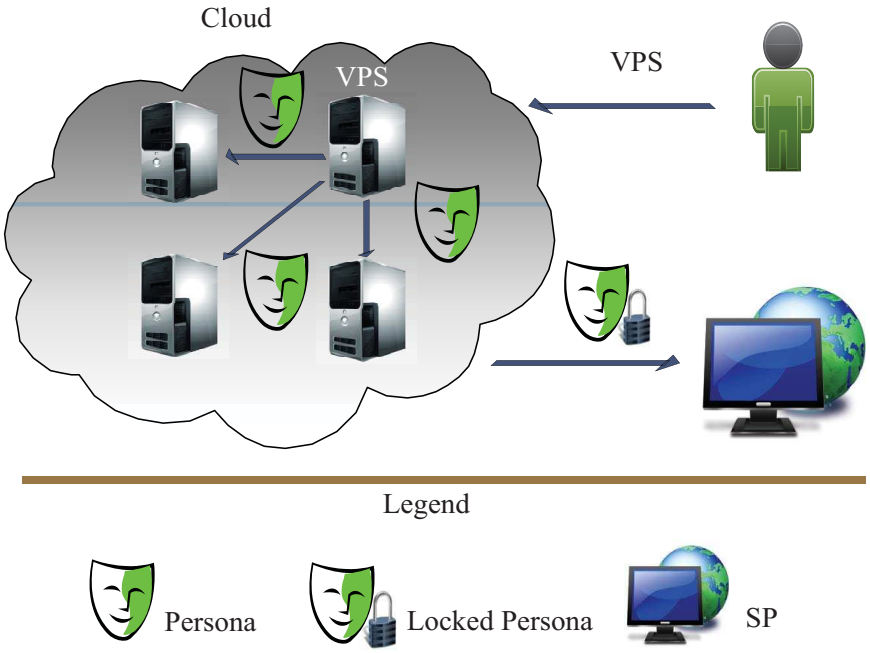

Figure 8.2: Cloud-based implementation of personas

- The individual removes the VPS from the cloud provider.

A VPS has the following properties.

- The individual may suspend, resume, or remove a VPS.
- Once a VPS is uploaded to a cloud provider, it distributes personas among the physical servers of that cloud, such that each persona is residing at a separate server.
- Each VPS has a public key pair which is used to encrypt personas before distributing them to the servers.
- A VPS does not decrypt a persona or use it, unless it receives an authenticated

message from the individual as follows. To use a persona, an individual contacts her VPS. The VPS authenticate the user. The authentication procedure between the individual and the VPS depends on initial setup of the VPS. It could be a username and a password, chosen by the individual, or a signed message, using a specific key.

The first property gives the individual control over her personas, and thus over her identity. The second property ensures that each persona is stored on a different server. If an attacker gains access to any of the cloud provider's servers, the attacker gains access to no more than one persona per individual. The third property is needed so that even if an attacker compromises a server, the attacker gets access to only encrypted personas. The fourth property protects personas from being decrypted without authenticating the individual requesting them.

8.2.1 Comparison to Current work

In the related work, individuals manage their certificates at their machines, which are limited in terms of computational and storage powers. In the presented approach, personas are managed at a cloud-computing environment.

In the related work, individuals are given certificates which they use to access resources at the clouds. This causes a problem, since all certificates of one individual are stored at her machine. If that machine is compromised by an attacker, all her certificates are compromised. In the presented approach, personas are distributed among a cloud's physical servers.

The related work assumes that individuals use web-anonymizers to hide individuals' IP address from service providers. In the presented approach, the cloud provider

which hosts a VPS acts as a web-anonymizer for the individual who uses that VPS.

8.2.2 System Design

Suppose that a persona provider uses *Wrap* to generate three personas, based on three attributes (A_1 to A_3), and sends them to an individual D. D receives the three personas (P_1 to P_3) and their respective secrets (S_1 to S_3).

$$\{P_1,\ S_1\} = Wrap(A_1),\ \{P_2,\ S_2\} = Wrap(A_2), \qquad \{P_3,\ S_3\} = Wrap(A_3)$$

D launches a VPS and supplies it with the three personas. The VPS generates a secret T for D to be used when D needs to use the personas. D sends the VPS to a cloud, as in Figure 8.2. The VPS encrypts and distributes the personas to three servers in the cloud.

Assume that D wants to interact with a service provider (SP), by signing a message M. D contacts the VPS and gets authenticated using T. D chooses which persona to use to interact with SP. The VPS contacts the servers where the chosen personas reside, and decrypts these personas. Each server then executes the *Show* operation to generate a locked persona using the decrypted persona it possesses. The servers send the locked persona to the SP.

Assume that P_2 is the only persona chosen. The server that possesses P_2 executes the *Show* operation to generate a locked persona L_M using P_2 and M. The locked persona is sent to the SP. The SP verifies the L_M and M with the *Verify* operation. Recall that each persona has a corresponding attribute A that the persona guarantees. In this case, A_2 is present when executing *Show* and *Verify*.

$$L_M = Show(M,\ P_2,\ S_2, A_2),\ \{true, false\} = Verify(M,\ L_M,\ A_2)$$

8.3 Chapter Summary

This chapter provides a cloud-based implementation of personas. This application has two advantages over current IMSs for cloud-computing environments. First, the computational and storage capabilities of the environment are utilized. Second, the credentials (personas) of an individual are distributed among n servers, where n is the number of these personas. This reduces the benefits that attackers gain by attacking a server.

Section 8.1 describes the background and related work. An IMS for cloud-computing environments is presented in Section 8.2. Section 8.3 summarizes the chapter.

Chapter 9

Conclusion

This thesis describes the drawbacks of using privacy policies to regulate the flow of identity information as an attempt to maintain individuals' privacy. The thesis shows how such strategy fails, especially with the advancement of data-mining and fusion techniques. These techniques threaten individuals' privacy by allowing organizations to fuse partial identities of individuals into profiles. Another major weakness of this strategy is that system administrators may configure policies in such a way that causes the release of identity information. Such incidents are reported in [3, 77].

Instead this thesis suggests the use of artificial identities, called personas, which can stand in for individuals in almost all circumstances, because they can be created with a full spectrum of properties, attributes, claims, desires, and relationships. Because these personas are, in a fundamental way, *single use* the potential for harm and loss of privacy is severely limited. The underlying properties can be achieved using cryptographic constructs.

Personas facilitate unlinkable interactions between an individual and an organization, as well as between a set of individuals and an organization. Personas are

based on cryptography constructs that are proven to be secure. The thesis presents a system for persona management. Then the design and implementation of the system are provided. The thesis assesses the threat model of the system.

Personas offer three important features, which constitute our contribution.

Encoding Relations

Personas may encode relations among individuals, allowing individuals to prove the relations to SPs, while preventing the SPs from profiling these individuals. In other words, the SP can verify the relationship among the individuals, yet if the individuals revisit the same SP, they cannot be distinguished from any other group of individuals having a relationship with the same structure. For instance, this feature is needed to allow a couple to prove the '*couple*' relationship to an SP.

Anonymity for SPs

There are many reasons for service providers to require anonymity. For example, some hotel chains use arbitrageurs to sell their room surplus at a lower price. Arbitrageurs hide the identity of the hotels until transactions are complete, so as not to undercut their full-price sales. Hotels cannot not reveal their brand names; therefore, they rely on arbitrageurs to prove hotel properties to customers. Customers cannot verify the offers from hotels; and thus, they have to trust the arbitrageurs.

Another example is anonymous double-auctions. In these auctions, buyers and sellers interact anonymously, and may change roles from buyers to sellers and vice versa. Personas allows service providers to be anonymous and permit customers to verify the qualities of the services, without reliance on arbitrageurs.

Constrained Interactions

Apart from individuals' identity attributes, service providers may need to limit

the rate at which individuals access services. For example, a reputation management system may permit an individual to submit no more than one reputation score for a given product, per time period. A provider may allow an individual to use a service n number of times per day.

We rely on public-key cryptography to facilitate the management of personas. In particular, personas are based on the identity-based signature scheme presented by Kiayias *et al.* [52]. The thesis presents the scheme and uses it to design cryptographic constructs to support personas.

A threat model is provided to describe the attacks that adversaries may launch against personas. Appendix A proves that personas are immune to the attacks suggested by the threat model.

The thesis applies personas in four areas. First, personas are used in e-commerce to allow service providers to interact anonymously with individuals. Personas help service providers to place anonymous offers, while help individuals to verify these offers. Thus, the necessity for arbitrageurs is relaxed. The application of personas in e-commerce is described in Chapter 5.

Second, personas is used to build an access-control model, the guarantee-based access control (GBAC). GBAC avoids the drawbacks of role-based access control (RBAC), such as the management of roles. GBAC allows fine-grained access policies to be specified. Relation-based access control is another feature of GBAC that allow access to be based on relations among individuals. The application of personas in access control is described in Chapter 6.

Third, personas are used in reputation-based trust management. Reputation management systems do not offer anonymity for individuals. Many anonymous reputation

management (ARM) systems allow for anonymity, but they fail to provide unlinkability of the anonymous ratings that an individual submits.

An ARM system for P2P networks is presented in this thesis. The transcripts of the interactions an individual has with a service provider are unlinkable. These transcripts can be used to enable individuals to submit ratings, without the fear of being profiled by the system. The ARM allows services to gain reputation, even if their customers neglected rating them. The system prevents Sybil attacks from degrading the quality of the reputation scores.

Moreover, the current work on privacy does not address trust management. Addressing privacy and trust in a single framework is more efficient, since the two subjects are related. In the Semantic Web [11] and the Semantic Social Web, trust is an integral part for automatic service discovery and invocation. Therefore, such a framework has a profound application in the Semantic-Web setting. The application of personas in reputation management is described in Chapter 7.

Fourth, the computational resources that cloud computing environments offer motivate the notion of cloud-based security. We use a cloud-based approach to enhance the privacy in identity-management systems. This is achieved by allowing individuals to move their personas to clouds that host personas. Individuals then can use their personas to interact with service providers. Since individuals need not store their personas at their client machines, attackers gain limited benefits when compromising these machines. Personas are stored in clouds in a distributed way to limit benefits of attacking these clouds. The application of personas in cloud-computing environments is presented in Chapter 8.

The limitations of personas are:

- Persona revocation is not implemented by the system. To compensate for this, the system changes its parameters, at specific time intervals, and re-issue personas for unrevoked individuals.

- Some systems allow for the evaluation of an access policy P, such that the evaluator does not know which set of identity attributes caused P to be satisfied. This minimizes the knowledge that the evaluator gain from evaluating P. This feature is not implemented by the system.

- If individuals share personas, then some individuals gain access rights to services they are not supposed to have. Personas do not implement techniques to deter individuals from sharing their personas.

- Personas are based on an identity-based signature scheme, which is based on pairing-based cryptography. Thus, personas take more time compared to systems that do not rely on pairing computations. However, research in number theory is enhancing the performance of pairing operations [8].

Bibliography

[1] 2008 digital future report final release highlights. Tech. rep., Center for the Digital Future, University of Southern California, California, US, 2008. Retrieved Jan 2010, from www.digitalcenter.org/pages/current_report.asp?intGlobalId=19.

[2] 2008 identity fraud survey report. Tech. report, Javelin Strategy and Research, Feb 2008. Retrieved Jan 2010, from www.idsafety.net/803.R_2008IdentityFraudSurveyReport_ConsumerVersion.pdf.

[3] Acohido, B. Hackers breach Heartland payment credit card system. USA Today, January 2009. Retrieved Jan 2010, from www.usatoday.com/money/perfi/credit/2009-01-20-heartland-credit-card-security-breach_N.htm.

[4] Al-Kahtani, M. A., and Sandhu, R. A model for attribute-based user-role assignment. In *Proceedings of the 18th Annual Computer Security Applications Conference* (2002), Las Vegas, NV, IEEE Computer Society, pp. 353–362.

[5] Amazon's Elastic Compute Cloud (EC2). Amazon. Retrieved Jan 2010, from aws.amazon.com/ec2.

[6] Artz, D., and Gil, Y. A survey of trust in computer science and the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web 5*, 2 (2007), Elsevier, 58–71.

[7] Backes, M., Camenisch, J., and Sommer, D. Anonymous yet accountable access control. In *Proceedings of the ACM Workshop on Privacy in the Electronic Society* (2005), Alexandria, VA, USA, ACM Press, pp. 40–46.

[8] Barreto, P., Galbraith, S., hÉigeartaigh, C., and Scott, M. Efficient pairing computation on supersingular abelian varieties. *Designs, Codes and Cryptography 42*, 3 (2007), Springer-Verlag, 239–271.

[9] MI6 boss in Facebook entry row. BBC News, July 2009. Retrieved Jan 2010, from `news.bbc.co.uk/2/hi/uk_news/8134807.stm`.

[10] Becker, M., and Sewell, P. Cassandra: Distributed access control policies with tunable expressiveness. In *Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks* (2004), Yorktown Heights, NY, IEEE Computer Society, pp. 159–168.

[11] Bemers-Lee, T., Hendler, J., and Lassila, O. The Semantic Web. *Scientific American 284*, 5 (2001), 34–43.

[12] Bertino, E., Bonatti, P. A., and Ferrari, E. TRBAC: A temporal role-based access control model. *ACM Transactions on Information and System Security 4*, 3 (2001), ACM Press, 191–233.

[13] Bertino, E., Paci, F., Ferrini, R., and Shang, N. Privacy-preserving digital identity management for cloud computing. *IEEE Data Engineering Bulletin 32*, 1 (2009), IEEE Computer Society, 21–27.

[14] Bhargav-Spantzel, A., Camenisch, J., Gross, T., and Sommer, D. User centricity: A taxonomy and open issues. *Journal of Computer Security 15*, 5 (2007), IOS Press, 493–527.

[15] Boneh, D., and Boyen, X. Short signatures without random oracles. In *Proceedings of the 24th International Conference on the Theory and Applications of Cryptographic Techniques* (2004), Interlaken, Switzerland, Springer-Verlag, pp. 56–73.

[16] Boneh, D., Boyen, X., and Shacham, H. Short group signatures. In *Proceedings of the 24th International Conference on the Theory and Applications of Cryptographic Techniques* (2004), Santa Barbara, CA, Springer-Verlag, pp. 41–55.

[17] Boneh, D., and Franklin, M. Identity-based encryption from the weil pairing. *SIAM Journal on Computing 32*, 3 (2003), Society for Industrial and Applied Mathematics, 586–615.

[18] Bramhall, P., Hansen, M., Rannenberg, K., and Roessler, T. User-centric identity management: New trends in standardization and regulation. *IEEE Security and Privacy 5*, 4 (2007), IEEE Computer Society, 84–87.

[19] Brin, S., and Page, L. The anatomy of a large-scale hypertextual web search engine. *Computer Network ISDN System 30*, 1-7 (1998), Elsevier, 107–117.

[20] Byun, J.-W., Bertino, E., and Li, N. Purpose based access control of complex data for privacy protection. In *Proceedings of the 10th ACM Symposium on Access Control Models and Technologies* (2005), Stockholm, Sweden, ACM Press, pp. 102–110.

[21] Cáceres, R., Cox, L., Lim, H., Shakimov, A., and Varshavsky, A. Virtual individual servers as privacy-preserving proxies for mobile devices. In *Proceedings of the 1st ACM Workshop on Networking, Systems, and Applications for Mobile Handhelds* (2009), Barcelona, Spain, ACM Press, pp. 37–42.

[22] Camenisch, J., and Herreweghen, E. V. Design and implementation of the Idemix anonymous credential system. In *Proceedings of the ACM Conference on Computer and Communications Security* (2002), Washington, DC, ACM Press, pp. 21–30.

[23] Camenisch, J., Shelat, A., Sommer, D., Fischer-Hübner, S., Hansen, M., Krasemann, H., Lacoste, G., Leenes, R., and Tseng, J. Privacy and identity management for everyone. In *Proceedings of the Workshop on Digital Identity Management* (2005), Fairfax, VA, ACM Press, pp. 20–27.

[24] Chappel, D. Introducing windows CardSpace. Microsoft, April 2006. Retrieved Jan 2010, from `www.msdn.microsoft.com/en-us/library/aa480189.aspx`.

[25] Chen, F., and Sandhu, R. S. Constraints for role-based access control. In *Proceedings of the first ACM Workshop on Role-based Access Control* (1996), Gaithersburg, MD, ACM Press, pp. 39–46.

[26] Creese, S., Hopkins, P., Pearson, S., and Shen, Y. Data protection-aware design for cloud computing. In *Proceedings of the first International Conference on Cloud Computing* (2009), Beijing, China, Springer-Verlag, pp. 119–130.

[27] D4.2: Set of requirements for interoperability of identity management systems. Tech. rep., FIDIS, 2006. Retrieved Jan 2010, from `www.fidis.net/resources/deliverables/interoperability`.

[28] Damiani, E., Vimercati, D. C. D., Paraboschi, S., Samarati, P., and Violante, F. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proceedings of the ACM Conference on Computer and Communications Security* (2002), ACM Press, pp. 207–216.

[29] Dimitriou, T., Karame, G., and Christou, I. Supertrust: A secure and efficient framework for handling trust in Super-Peer networks. In *Proceedings of the 26th annual ACM Symposium on Principles of Distributed Computing* (2007), Portland, OR, ACM Press, pp. 374–375.

[30] Dingledine, R., Mathewson, N., and Syverson, P. TOR: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium* (2004), San Diego, CA, USENIX Association, pp. 21–21.

[31] Douceur, J. The Sybil attack. In *Proceedings of the first International Workshop on Peer-to-Peer Systems* (2002), Cambridge, MA, pp. 251–260.

[32] Fiat, A., and Shamir, A. How to prove yourself: Practical solutions to identification and signature problems. In *Proceedings of the Conference on Advances in Cryptology* (1986), Santa Barbara, CA, Springer-Verlag, pp. 186–194.

[33] Foster, I., Zhao, Y., Raicu, I., and Lu, S. Cloud computing and grid computing 360-degree compared. In *Proceedings of the Grid Computing Environments Workshop* (2008), IEEE Computer Society, pp. 1–10.

[34] Frankowski, D., Cosley, D., Sen, S., Terveen, L., and Riedl, J. You are what you say: privacy risks of public mentions. In *Proceedings of the International ACM Conference on Research and Development in Information Retrieval* (2006), Seattle, WA, ACM Press, pp. 565–572.

[35] Gartner press releases. Gartner, July 2008. Retrieved Jan 2010, from `www.gartner.com/it/page.jsp?id=722307`.

[36] Golbeck, J., and Hendler, J. Accuracy of metrics for inferring trust and reputation in Semantic Web-based social networks. In *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management* (2004), Whittlebury, UK, pp. 116–131.

[37] Goldschlag, D., Reed, M., and Syverson, P. Onion routing. *Communications of the ACM 42*, 2 (1999), ACM Press, 39–41.

[38] Goldwasser, S., Micali, S., and Rackoff, C. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing 18*, 1 (1989), Society for Industrial and Applied Mathematics, 186–208.

[39] Google App Engine. Retrieved Jan 2010, from `code.google.com/appengine`.

[40] Grandison, T., and Sloman, M. A survey of trust in Internet applications. *IEEE Communications Surveys and Tutorials 3*, 4 (2000), IEEE Communications Society, 2–16.

[41] Hansen, M., Berlich, P., Camenisch, J., Clauss, S., Pfitzmann, A., and Waidner, M. Privacy-enhancing identity management. *Information Security Technical Report 9*, 1 (2004), Elsevier, 35–44.

[42] Hayat, A. *A Pan European Interoperable Electronic Identity.* Ph.D. thesis, Graz University of Technology, Austria, 2007.

[43] How cloud computing is changing the world. Business Week, August 2008. Retrieved Jan 2010, from `www.businessweek.com/technology/content/aug2008/tc2008082_445669.htm`.

[44] Hussain, M., and Skillicorn, D. B. Persona-based identity management: A novel approach to privacy protection. In *Proceedings of the 13th Nordic Workshop on Secure IT Systems* (2008), Copenhagen, Denmark, Technical University of Denmark, pp. 201–212.

[45] Hussain, M., and Skillicorn, D. B. Guarantee-based access control. In *Proceedings of the IEEE International Conference on Computational Science and Engineering* (2009), Vancouver, Canada, IEEE Computer Society, pp. 201–206.

[46] Identity fraud and identity theft, the uks fraud prevention service. Retrieved Jan 2010, from `www.cifas.org.uk/default.asp?edit_id=561-56`.

[47] Identity, privacy and the need of others to know who you are: A discussion paper on identity issues. Tech. report, Office of the Privacy Commissioner of Canada,

September 2007. Retrieved Jan 2010, from `www.privcom.gc.ca/information/pub/id_paper_e.pdf`.

[48] Josang, A., Zomai, M. A., and Suriadi, S. Usability and privacy in identity management architectures. In *Proceedings of the Fifth Australasian Symposium on ACSW frontiers* (2007), Ballarat, Australia, Australian Computer Society, pp. 143–152.

[49] Kagal, L., Finin, T., and Joshi, A. Developing secure agent systems using delegation based trust management. In *Proceedings of Security of Mobile Multi-Agent Systems Workshop, held at Autonomous Agents and MultiAgent Systems Conference* (2002), Bologna, Italy, ACM Press, pp. 27–34.

[50] Kagal, L., Finin, T., and Joshi, A. A policy based approach to security for the Semantic Web. In *Proceedings of the International Semantic Web Conference* (2003), Sanibel Island, FL, Springer-Verlag, pp. 402–418.

[51] Kamvar, S., Schlosser, M., and Garcia-Molina, H. The Eigentrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th International Conference on World Wide Web* (2003), Budapest, Hungary, ACM Press, pp. 640–651.

[52] Kiayias, A., and Zhou, H.-S. Hidden identity-based signatures. In *Proceedings of the 11th International Conference on Financial Cryptography and Data Security* (2008), Scarborough, Trinidad and Tobago, Springer-Verlag, pp. 134–147.

[53] Kumar, A., Karnik, N., and Chafle, G. Context sensitivity in role-based access control. *SIGOPS Operating Systems Review 36*, 3 (2002), ACM Press, 53–66.

[54] Levy, H. M. *Capability-Based Computer Systems.* Butterworth-Heinemann, Newton, MA, US, 1984.

[55] Liberty Alliance Project specifications. Retrieved Jan 2010, from `www.projectliberty.org/liberty/specifications__1`.

[56] Lynn, B. Pairing-based Cryptography Library, Retrieved Jan 2010, from `crypto.stanford.edu/pbc`.

[57] Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. L-diversity: Privacy beyond K-anonymity. *ACM Transactions on Knowledge Discovery from Data 1*, 1 (2007), ACM Press, 1–52.

[58] Madsen, P., and Itoh, H. Challenges to supporting federated assurance. *Computer 42*, 5 (2009), IEEE Computer Society, 42–49.

[59] Malin, B., and Sweeney, L. How (not) to protect genomic data privacy in a distributed network: using trail re-identification to evaluate and design anonymity protection systems. *Journal of Biomedical Informatics 37*, 3 (2004), Elsevier Science, 179–192.

[60] McGuinness, D., and Harmelen, F. V. OWL web ontology language overview, February 2004. W3C Recommendation, Retrieved Jan 2010, from `www.w3.org/TR/owl-features`.

[61] Miyata, T., Koga, Y., Madsen, P., Adachi, S.-I., Tsuchiya, Y., Sakamoto, Y., and Takahashi, K. A survey on identity management protocols and standards. *Transactions on Information and Systems E89-D*, 1 (2006), Oxford University Press, 112–123.

[62] Müller, W., Plötz, H., Redlich, J.-P., and Shiraki, T. Sybil proof anonymous reputation management. In *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks* (2008), Istanbul, Turkey, ACM Press, pp. 1–10.

[63] Muttik, I., and Barton, C. Cloud security technologies. *Information Security Technical Report 14*, 1 (2009), Elsevier, 1–6.

[64] Narayanan, A., and Shmatikov, V. Robust de-anonymization of large sparse datasets. In *Proceedings of the IEEE Symposium on Security and Privacy* (2008), Oakland, CA, IEEE Computer Society, pp. 111–125.

[65] Oppliger, R. Microsoft .NET Passport: A security analysis. *Computer 36*, 7 (2003), IEEE Computer Society, 29–35.

[66] Paci, F., Ferrini, R., Musci, A., and Steuer, K.and Bertino, E. An interoperable approach to multifactor identity verification. *Computer 42*, 5 (2009), IEEE Computer Society, 50–57.

[67] Peterson, G. Introduction to identity management risk metrics. *IEEE Security and Privacy Magazine 4*, 4 (2006), IEEE Computer Society, 88.

[68] Recordon, D., and Reed, D. Openid 2.0: A platform for user-centric identity management. In *Proceedings of the Second ACM Workshop on Digital Identity Management* (2006), Alexandria, VI, ACM Press, pp. 11–16.

[69] Rezgui, A., Bouguettaya, A., and Malik, Z. A Reputation-based approach to preserving privacy in Web services. *Lecture Notes in Computer Science 2819* (2003), Springer, 91–103.

[70] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E. Role-based access control models. *IEEE Computer 29*, 2 (1996), IEEE Computer Society, 38–47.

[71] Security Assertion Markup Language (SAML). Retrieved Jan 2010, from `saml.xml.org`.

[72] Sekhavat, Y., and Fathian, M. Efficient anonymous secure auction schema(ASAS) without fully trustworthy auctioneer. *Information Management & Computer Security 16*, 3 (2008), Emerald, 288–304.

[73] Shamir, A. Identity-based cryptosystems and signature schemes. In *Proceedings of the International Cryptology Conference on Advances in Cryptology* (1985), Santa Barbara, CA, Springer-Verlag, pp. 47–53.

[74] Shibboleth. Retrieved Jan 2010, from `www.shibboleth.internet2.edu`.

[75] Singh, A., and Liu, L. TrustMe: Anonymous management of trust relationships in decentralized P2P systems. In *Proceedings of the third Conference on Peer-to-Peer Computing* (2003), Linkoping, Sweden, IEEE Computer Society, pp. 142–149.

[76] Skillicorn, D. B., and Hussain, M. Personas: Beyond identity protection by information control. Tech. report, Queen's University, Kingston, Canada, March 2009. Commissioned by the Office of the Privacy Commissioner of Canada, Retrieved Jan 2010, from `research.cs.queensu.ca/home/skill/opccreport.pdf`.

[77] Sullivan, B. 40 million credit cards exposed. MSNBC, June 2005. Retrieved Jan 2010, from `www.msnbc.msn.com/id/8260050`.

[78] Sweeney, L. K-anonymity: A model for protecting privacy. *International Journal of Uncertainty Fuzziness and Knowledge Based Systems 10*, 5 (2002), World Scientific Singapore, 557–570.

[79] Sxip Inc. Retrieved Jan 2010, from `www.sxip.com`.

[80] The Higgins Identity Framework. Retrieved Jan 2010, from `www.eclipse.org/higgins`.

[81] Trevathan, J., Ghodosi, H., and Read, W. An anonymous and secure continuous double auction scheme. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences* (2006), vol. 6, Kauai, HI, IEEE Computer Society, p. 125b.

[82] U-Prove SDK overview. white paper, Credentica Inc, 2007. Retrieved Jan 2010, from `www.credentica.com/files/U-ProveSDKWhitepaper.pdf`.

[83] Web Services Federation language. Retrieved Jan 2010, from `www.ibm.com/developerworks/library/specification/ws-fed`.

[84] Yan, L., Rong, C., and Zhao, G. Strengthen Cloud Computing Security with Federal Identity Management Using Hierarchical Identity-Based Cryptography. In *Proceedings of the first International Conference on Cloud Computing* (2009), Beijing, China, Springer-Verlag, pp. 167–177.

[85] Yu, T., Winslett, M., and Seamons, K. Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation. *ACM Transactions on Information and System Security 6*, 1 (2003), ACM Press, 1–42.

[86] Yuan, E., and Tong, J. Attributed based access control (ABAC) for web services. In *Proceedings of the IEEE International Conference on Web Services* (2005), Orlando, FL, IEEE Computer Society, pp. 561–569.

Appendix A

Correctness and Security

The threat model described in Section 3.4 suggests several attacks: forging a persona, forging a locked persona, forging an attribute, linking locked personas, and de-anonymizing a locked persona. Below is the list of the attacks and their prerequisites.

- *Forging personas.* Forging personas can be achieved if the *Wrap* operation is insecure. *Rationale.* If *Wrap* is not secure, an attacker may issue personas valid with respect to a PP, without having the private key of the PP.

- *Forging locked personas or attributes.* Forging locked personas or attributes can be achieved if the *Wrap* or the *Show* operations are insecure. *Rationale.* If *Wrap* is not secure, an attacker may issue personas, and thus, may generate valid locked personas. If *Show* is not secure, an attacker may use a valid persona to generate locked personas with attributes that the attacker is not entitled to.

- *Linking or de-anonymizing locked personas.* Linking or de-anonymizing locked personas can be achieved if the *Trace* operation is insecure. *Rationale.* If *Trace*

is not secure, an attacker may de-anonymize a set of locked personas and link those which belongs to the same persona to each other.

To protect the system against the attacks described in the threat model, we need to make sure that the operations that manage personas are correct, as well as secure. The following sections prove the correctness and security of the operations.

The operations provided by the system can be categorized into: secure interaction operations and reputation-management operations. The first category includes: *Wrap*, *Show*, *Verify*, *Trace*, *SelectiveShow* and *VerifyRelation*. The second includes: *SubmitReputation*, *VerifyReputation*, *UpdateReputation*, *GenerateTicket*, and *VerifyTicket*. The correctness and security analysis is structured according to the categories.

A.1 Correctness of Secure Interaction Operations

The correctness and security of the secure interaction operations is mainly drawn from the correctness and security of the underlying hidden ID-based signatures. *Wrap*, *Show*, *Verify*, and *Trace* are the operations in which other secure interaction operations are built on-top. Proving the correctness and security of *Wrap*, *Show*, *Verify* and *Trace* operations implies the security and correctness of the remaining ones. The operations are correct if the following three conditions hold. The probability that *Check_Wrap* evaluates to true is 1, given that it is executed on a valid (persona, secret) pair, generated by *Wrap*.

$$Probability[true \leftarrow Check\_Wrap(persona, secret, PP\ pparam) \mid$$
$$(persona, secret) \leftarrow Wrap(attributes, proof, PP\ pparam, PP\ prkey)] = 1$$

Verify always succeeds when executed on a valid locked persona *lpersona*.

$$
\begin{aligned}
&\mathit{Probability}[\\
&\quad \mathit{true} \leftarrow \mathit{Verify}(\mathit{lpersona}, \mathit{PP\ pparam}, \mathit{DA\ pparam}) \mid \\
&\quad (\mathit{persona}, \mathit{secret}) \leftarrow \mathit{Wrap}(\mathit{attributes}, \mathit{proof}, \mathit{PP\ pparam}, \mathit{PP\ prkey});\\
&\quad \mathit{true} \leftarrow \mathit{Check\_Wrap}(\mathit{persona}, \mathit{secret}, \mathit{PP\ pparam});\\
&\quad \mathit{lpersona} \leftarrow \mathit{Show}(\mathit{persona}, \mathit{secret}, \mathit{PP\ pparam}, \mathit{DA\ pparam})] = 1
\end{aligned}
$$

Trace always extracts the persona used by the show operation to generate a locked persona verifiable by the verify operation.

$$
\begin{aligned}
&\mathit{Probability}[\\
&\quad \mathit{persona} \leftarrow \mathit{Trace}(\mathit{lpersona}, \mathit{DA\ pparam}, \mathit{DA\ prkey}) \mid \\
&\quad (\mathit{persona}, \mathit{secret}) \leftarrow \mathit{Wrap}(\mathit{attributes}, \mathit{proof}, \mathit{PP\ pparam}, \mathit{PP\ prkey});\\
&\quad \mathit{true} \leftarrow \mathit{Check\_Wrap}(\mathit{persona}, \mathit{secret}, \mathit{PP\ pparam});\\
&\quad \mathit{lpersona} \leftarrow \mathit{Show}(\mathit{persona}, \mathit{secret}, \mathit{PP\ pparam}, \mathit{DA\ pparam});\\
&\quad \mathit{true} \leftarrow \mathit{Verify}(\mathit{lpersona}, \mathit{PP\ pparam}, \mathit{DA\ pparam})] = 1
\end{aligned}
$$

The proof of correctness of the hidden ID-based signature scheme is presented by Kiayias *et al.* [52]. *Wrap* is implemented by $\mathit{Register}_e$, which is a composition of two instances of *Register* in HIDS. *Show* is implemented by Show_e, which is a composition of two instances of *Sign* in HIDS. *Verify* is implemented by Verify_e, which is a composition of two instances of *Verify* in HIDS. *Trace* is implemented by Open_e, which is a composition of two instances of *Open* in HIDS. Therefore, the three conditions described above hold. *Wrap*, *Show*, *Verify*, and *Trace* are correct, based

on the correctness of the operations in the HIDS scheme.

SelectiveShow is a composition of n instances of $Sign_e$, where n is the number of different sets of attributes the individual wishes to prove. *VerifyRelation* is a composition of two instances of $Verify_e$, plus an additional check for the relation $(W_1\, g^{relation} = W_2)$, where $W_1 = w^{l+k}\, g^{I_1}$, and $W_2 = w^{l+k}\, g^{I_2}$. Clearly, the check holds only if $relation = I_2 - I_1$.

Recall that when a PP validates the relation between two individuals D_1 and D_2, the PP makes the pseudonym part of their identifiers to be equal. The PP also makes the difference between relation part of their identifiers to be *relation*. Thus, *relation* is equal to $I_2 - I_1$, only if the PP did certify the relation between D_1 and D_2. Therefore, *SelectiveShow* and *VerifyRelation* are correct based on the correction of *Wrap*, *Show*, *Verify*, and *Trace*.

Let Algorithm 4.1 and 4.2 are used to generate personas and locked personas to prove relations among a group of n individuals. We prove the correctness of *VerifyRelation* when it is used by a group of individuals. We begin by the simple case, which is to prove that the first individual relations with other individuals are verified by the first call to *VerifyRelation* Algorithm 4.2. At the first call to *VerifyRelation*, the following holds: $relation = \sqrt{(R_1^1)^2 + (R_1^2)^2 + ... + (R_1^n)^2}$, the relation part of the locked persona LP_1 is equal to 0 (follows from Algorithm 4.1), and the relation part of the locked persona LP_2 is equal to $\sqrt{(R_1^1)^2 + (R_1^2)^2 + ... + (R_1^n)^2}$ (follows from Algorithm 4.1). The difference between the relation parts of LP_1 and LP_2 is equal to the value of the relations of the first individual with the rest of the individuals. Therefore, the relations of the first individual with other individuals are verified by the first call to *VerifyRelation*.

Now we take the general case, which is proving that the k^{th} call to *VerifyRelation* verifies the relations of the k^{th} individual with the rest of the individuals. At the k^{th} call, let the relation part of the k^{th} locked persona (LP_k) is equal to Q. The relation part of LP_{k+1} is equal to $E = Q + \sqrt{(R_k^1)^2 + (R_k^2)^2 + ... + (R_k^n)^2}$ (from Algorithm 4.1). The difference between the relation parts of LP_k and LP_{k+1} is equal to $\sqrt{(R_k^1)^2 + (R_k^2)^2 + ... + (R_k^n)^2}$, which is the encoding of the relations of the k^{th} individual. Therefore, the k^{th} call to *VerifyRelation* verifies the relations of the k^{th} individual.

A.2 Correctness of Reputation-Management Operations

SubmitReputation and *VerifyReputation* are correct if the following condition holds. *VerifyReputation* always succeeds when executed on a valid reputation message. Since the implementation of *UpdateReputation* varies from one domain into another, and is based on system administrators, the correctness of the operation not discussed.

$$
\begin{aligned}
&Probability[\\
&\quad true \leftarrow VerifyReputation(reputation, lpersona_r, DA\ pparam, PP\ pparam)\ |\\
&\quad lpersona_i \leftarrow Show(persona, secret, PP\ pparam, DA\ pparam);\\
&\quad true \leftarrow Verify(lpersona_i, PP\ pparam, DA\ pparam);\\
&\quad reputation, lpersona_r \leftarrow SubmitReputation(score, lpersona_i, persona,\\
&secret, DA\ pparam, PP\ pparam, SP, response)] = 1
\end{aligned}
$$

SubmitReputation generates a locked persona on a reputation message. *SubmitReputation* uses *Show* to generate the locked persona. *VerifyReputation* verifies the locked persona contained in the reputation message $lpersona_i$, and the locked persona generated on the reputation message $lpersona_r$. *VerifyReputation* uses *Verify* to verify the locked personas. *VerifyReputation* uses the RSA public key of SP to validate the SP's response contained in the reputation message, and to make sure that the individual had an interaction with the SP being rated.

Since *Show* and *Verify* are proven to be correct, and the RSA public cryptography is correct, then the above condition holds. *SubmitReputation* and *VerifyReputation* are correct, based on the correctness of *Show*, *Verify* and RSA.

GenerateTicket and *VerifyTicket* are correct if the following condition holds.

$$
\begin{aligned}
&Probability[\\
&\quad true \leftarrow VerifyTicket(ticket, tRequest, PP\ pparam)\ |\\
&\quad ticket \leftarrow GenerateTicket(tRequest, PP\ prkey, PP\ pparam, response)] = 1
\end{aligned}
$$

Recall that a ticket request *tRequest* consists of two hashed strings, generated by a hash function. *GenerateTicket* sign ticket requests to generate tickets. Tickets are RSA signatures on a ticket request. *VerifyTicket* uses signature verification of RSA to verify tickets against ticket requests. Since RSA signatures generated by RSA signing keys are verified by the corresponding RSA verification keys, the above condition holds. Therefore, *GenerateTicket* and *VerifyTicket* are correct, based on the correctness of RSA public cryptography.

A.3 Security of Secure Interaction Operations

Wrap, *Show*, *Verify*, and *Trace* are secure against misidentification attacks, if the probability of an adversary succeeding in the following game is negligible. In this game, the adversary has access to *Wrap_Oracle*, which executes *Wrap* and returns the resultant (persona, secret) pair. The adversary has access to *Show_Oracle*, which executes *Show* and returns the resultant locked persona. The adversary wins the game if it produces a valid locked persona that is untraceable to a persona. It also wins if it generates a traceable locked persona, but without using *Wrap_Oracle* to receive the persona associated with that locked persona, and without using *Show_Oracle* to produce that locked persona.

Wrap_Oracle(*attributes*)

$(persona, secret) \leftarrow Wrap(attributes, proof, PP\ pparam, PP\ prkey)$

$Personas \leftarrow \{persona\} \cup Personas$

return $(persona, secret)$

Show_Oracle(*persona*)

$lpersona \leftarrow Show(persona, secret, PP\ pparam, DA\ pparam)$

$Lpersona \leftarrow \{lpersona\} \cup Lpersonas$

return $lpersona$

Misidentification_Game()

lpersona $\leftarrow$ *Adversary*(*Wrap_Oracle*, *Show_Oracle*)

if(*true* $\leftarrow$ *Verify*(*lpersona*, *PP pparam*, *DA pparam*) **AND**

$\Phi \leftarrow$ *Trace*(*lpersona*, *DA pparam*, *DA prkey*))

Adversary wins

elseif(*true* $\leftarrow$ *Verify*(*lpersona*, *PP pparam*, *DA pparam*) **AND**

persona $\leftarrow$ *Trace*(*lpersona*, *DA pparam*, *DA prkey*) **AND**

persona $\notin$ *Personas* **AND** *lpersona* $\notin$ *Lpersonas*)

Adversary wins

else *Adversary loses*

Wrap, *Show*, *Verify*, and *Trace* are secure against adaptive chosen-cyphertext attacks (CCA2), if the probability of an adversary succeeding in the following game is $0.5 + \epsilon$, where ϵ is negligible. The adversary has access to *Trace_Oracle*, which reveals the persona used to generate a locked persona. The adversary is presented with a locked persona and two personas, in which one persona was used to generate the locked persona. The adversary wins the game if it guesses the right persona. Of course, the adversary is constrained from using *Trace_Oracle* on the presented locked persona. *Trace_Oracle*$_\alpha$ refers to that constraint.

Trace_Oracle($lpersona$)

$persona \leftarrow Trace(lpersona, DA\ pparam, DA\ prkey)$

$return \quad persona$

CCA2_Game()

$(persona_1, secret_1) \leftarrow Wrap(attributes_1, proof_1, PP\ pparam, PP\ prkey)$

$(persona_2, secret_2) \leftarrow Wrap(attributes_2, proof_2, PP\ pparam, PP\ prkey)$

$r \leftarrow random \quad from \quad \{1, 2\}$

$lpersona \leftarrow Show(persona_r, secret_r, PP\ pparam, DA\ pparam)$

$challenge \leftarrow \{lpersona, persona_1, persona_2\}$

$guess \leftarrow Adversary(trace\_oracle_\alpha, challenge)$

if($guess = persona_r$)

$Adversary \quad wins$

else $Adversary \quad loses$

Suppose that an adversary A has the ability to launch successful misidentification and / or CCA2 attacks on *Wrap*, *Show*, *Verify*, and *Trace*. Those operations are implemented by $Register_e$, $Sign_e$, $Verify_e$, and $Open_e$ in extended HIDS, respectively, whereas $Register_e$, $Sign_e$, $Verify_e$, and $Open_e$ are instances of HIDS operations. Therefore, A can launch successful misidentification and / or CCA2 attacks on the HIDS operations.

The HIDS scheme is proven to be secure against misidentification and CCA2

attacks under the Strong Diffie Hellman (SDH) [15] and Decisional Linear Diffie Hellman (DLDH) [16] assumptions in the random oracle model. The security proof is presented by Kiayias *et al.* [52]. Since HIDS operations are proven to be secure, then *Wrap*, *Show*, *Verify*, and *Trace* are also secure against misidentification and / or CCA2 under the SDH and DLDH assumptions, in the random oracle model. The security of *SelectiveShow* and *VerifyRelation* follows from the security of *Wrap*, *Show*, *Verify*, and *Trace*. This is because *SelectiveShow* and *VerifyRelation* are composed of several instances of *Show* and *Verify* operations.

A.4 Security of Reputation-Management Operations

Show and *Verify* are secure, and RSA public cryptography is secure. Therefore, *SubmitReputation* and *VerifyReputation* are secure. The security of *GenerateTicket* and *VerifyTicket* follows from the security of the RSA public cryptography.